BEEBLET

ISSN 0112-3351

VOLUME 2 NUMBER 11 DECEMBER 1984 / JANUARY 1985

BBC/ACORN
COMPUTER USER GROUP
NZ (inc)

CONTENTS

Contents 2
Editorial 3
Around the regions 5
BBC Starter Pack -Review 6
Cartoon 7
Initiate Your Grandparents 8
Hyper Hopper 12
12 Days of Beebmas 14
Get Rid of that Pain in the Neck 15
Vampire Adventure - Review 16
Electron 17
GORF -Review 18
Toolkit -Review 19
Elite -Review 21
1985 Calender 23
Text Display Speed 25
Left Right Frogs 27
Zapper 33
Enigma Solution 35
Mental Arithmetic Game 37
Table of Overlaps 39
OSWORD from BASIC! 41
Edit Characters II 43
Co-Processor Screen Dump 44
A Hex on that Character 45
Display Character Marix 48
Crossword 49
Graphs 53
Move to Print 60
Wanted, Not Wanted, Help 61
Cursor Help 62
Mailbag 64
Random Palette 66
Beepers 67
Sideways ROM 68
Screen Dump in BASIC 68
Invaluable Utilities for the BBC Micro 70
Thanks for the Memory 73
Classified Ads 77
Hall of Fame 79
Local Groups 80

ADVERTISERS

APTEC. 20
Computer Centre 63
Einstein Scientific 72
Penguin Books 36
Pitman Publishing 4
Roberts Print 78

BEEBLET is edited by Bruce Wills who may sometimes be tracked down on 861-662 (h) or 720-030 (wk).

DISCLAIMER

The views of the reviewers, the Editor and other contributors are not necessarily shared by the BBC/ACORN COMPUTER USER GROUP OF NZ INC.

EDITORIAL

Merry Christmas and a happy new year to all. I would especially like to *thank all the regular contributors, and in particular Anne and* Gavin Flower, for the tremendous amount of creative energy and support they provided. Many others sent in splendid contributions ranging from a few lines to full blown articles. It is on these that Beeblet is built.

Of course, the break will give you all that little bit of time you *need to finish that contribution you've been meaning to send in for* Beeblet all year. Won't it? This year seems to have sort of snuk past on me: ever since I took over this mag (nominally anyway) all my other interests / commitments seemed to have expanded at a geometric rate as the year progressed - that, and the silly season, is why no one could get me at home in the evenings in the fortnight or so *preceding Beeb production weekend.*

****

A.G.M.

The next Beeblet will come out in February (early February if I get enough material) so I will remind you now to think about things you would like to see on the A.G.M. agenda.

****

ATTENTION TEACHERS

A lot of fuss was made about concessions for students, but as far as I know only one school has taken up the special student membership concession which was approved at the last AGM.

****

COMMUNICATIONS

I know that the various branches around the country are organising workshops, seminars etc. Eg. Christchurch has what sounds like an interesting series on Robotics. Could someone write these up for the mag? The exchange of knowledge and ideas is after all the basis of our club.

****

SERVICING

As the Aptec add in this issue suggests we have a new arrival on the Wellington BBC servicing scene, strongly endorsed by Barry S. of Lower Hutt. Aptec are not an accredited BBC service agent at this stage, but may seek accreditation if demand warrants it.

****

COMPETITION

The only thing approaching an entry in the create an educational program competition was G.C.F.'s mental arithmetic. I guess it wins!

****

THE END

Well, that's how stories are supposed to end. Following this effort, my house is in a state of total chaos, my body is not far behind, and even the C-Itoh ground to halt for no apparent reason this morning (mind you, like me it has been running almost continuously over the last 72 hours...and its only a machine...). Enjoy the break. I will!

NEW BBC SOFTWARE PROGRAMS
from

Pitman Publishing

GAME CORE

This intriguing software gives you all the information you need to write simple and highly absorbing board games for your computer. It includes three ready made games and a fourth is explained in detail as a tutorial model. There is an introduction to game trees and strategies, an explanation of coding, and guidance on writing your own games programs.

$49.00

BBC Model B Cassette Version
ISBN 0 563 21072 9

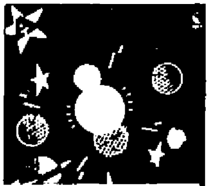
DRAWSTICK

A comprehensive yet friendly and versatile drawing program which can produce complex high-resolution pictures in up to four colours, with the ability to save all options to disc and tape allowing later modifications. It enables children aged from 8 upwards to create pictures, charts, maps and diagrams, and is sufficiently sophisticated for use by older children or adults as a Computer Aided Design tool.

$45.00

BBC Model B Cassette Version
ISBN 0 563 21086 9

ASTRONOMY



Children of all ages are fascinated by space and astronomy. This software has a range of vivid graphical demonstrations which bring astronomical subjects to life. Aimed at the 9-12 age range the programs include Phases of the Moon, Eclipses, Seasons of the Year, Planetary Motions, and Lander. The first four programs are demonstration models of common astronomical phenomena shown in animated colour graphics. Lander is a gravitationally accurate game where a spacecraft has to be landed on several different planets.

BBC Model B Disc Version $58.00
ISBN 0 563 32650 6

BBC Model B Cassette Version $40.00
ISBN 0 563 32651 4

WHITE KNIGHT MARK II (ELECTRON)

The amazing chess program that took joint first place in the home computer section of the PEW European Chess Championships, now versioned for Electron.

$45.00

Acorn Electron Version
ISBN 0 563 21096 6

Further information available from

PITMAN PUBLISHING NEW ZEALAND LIMITED
PO BOX 38-688
28 FITZHERBERT STREET
PETONE

AROUND THE REGIONS

DUNEDIN

Martin McDowall reports that a group is slowing developing in Dunedin, through Whitcoulls. Give me contact names and numbers and I will list them on the back cover.

INVERCARGILL

Evan Cheyne reported last month that a local group was being established. Advantages of joining include:
(a) Meeting other people with a common interest.
(b) Solving problems you may be having with your equipment.
(c) DEALS. - but NOT a swap session for pirated software.

CHRISTCHURCH 1985 PROGRAM:

4 Feb	Introductory class
18 Feb	" "
4 March	VIEW word processing
18 March	Discs & Filing
1 April	Open Forum
15 April	Sound & Graphics
29 April	Fundamentals of Assembly Language
27 May	Good Programming Techniques I
10 June	" " " II
24 June	Guest speaker T.B.A.
8 July	Open forum

Meetings every 2nd Monday 7pm at Hagley H.S.

TOKOROA

A new Users Group! Meeting first Friday of each Month, 8pm at Tokoroa High School. Contact Brian ph. 67625

NIGHT CLASSES next year there too:
8 week basic computer studies;
8 week intermediate computer studies
Each class 2 hours, each course $16.00
Enquiries Brian Mycroft, P.O. Box 128, Tokoroa

BBC STARTER PACK

Collins Software, $55-ish. Review copy supplied by Whitcoulls

Reviewed by Anne F.

"Someone should write an program for people to learn how to write programs".

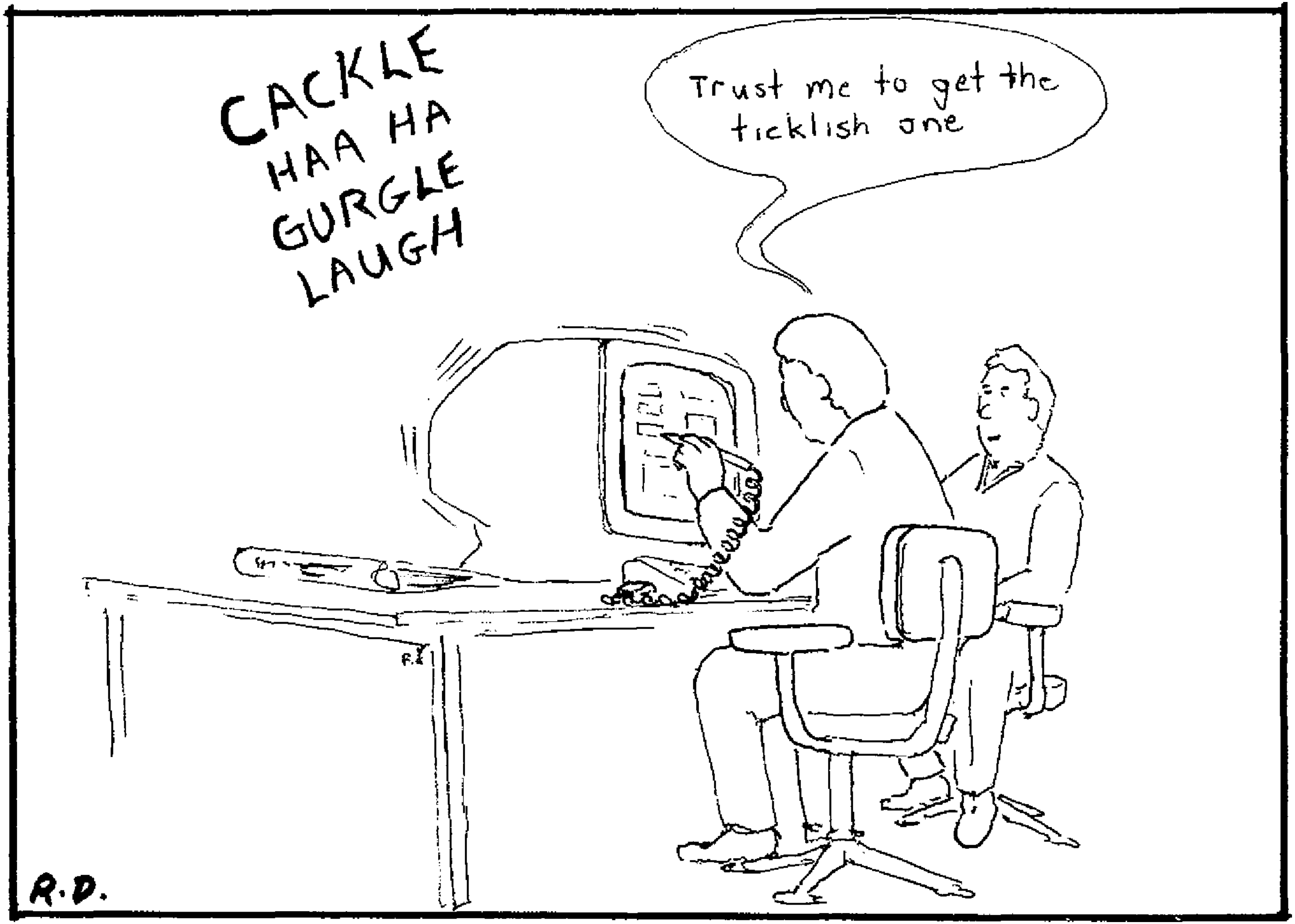
Well, here it is. The Starter Pack is a 60-page book and two cassettes in a vinyl folder. It is aimed at the complete beginner, to teach them to get the computer obeying simple instructions, and leads them to slightly greater complexities.

The package is divided into 10 lessons; each lesson is a book chapter and a program on tape. The user is expected to have the book alongside the computer, and the keyboard only accepts the answer wanted by the program. So you cannot type in "RUN" if it is wanting "NEW" or "LIST" or Space Bar! You write programs, edit them etc as you normally would, BUT you are still within the chapter program! It is a good approach for beginners afraid of making a mistake - an incorrect response often doesn't even appear on the screen!

Most of the programs are in Mode 7 and use colour, but illustrative graphics (other than Collins' house logo on the title screen) are left to the last chapter (and barely mentioned there). The pack could easily be re-written for another machine - not a PROC in sight, and Chapter 9 suggests that although GOTO is a powerful command, overuse can make your program difficult to follow and modify! PROCs are too advanced for this book, as are string handling, colour and graphics, sound...

A beginner needn't then feel overwhelmed by the amount expected to learn. The lessons expect you to 'dry run', or predict the result of, each program and its variations, and the book has spaces for the result to be written before actually RUNning each program. This is a good approach, and should get beginners understanding what a computer is capable of doing, and how they can get simple programs written and working properly before attacking the keyboard. It is a well-thought-out teaching package. Its main fault was trying to load block 24 of the third program! On either side of the tape!

Q. When is a number not a number, but a string?
A. When it's a house number.


CACKLE
HAA HA
GURGLE
LAUGH
Trust me to get the ticklish one
R.D.

INITIATE YOUR GRANDPARENTS

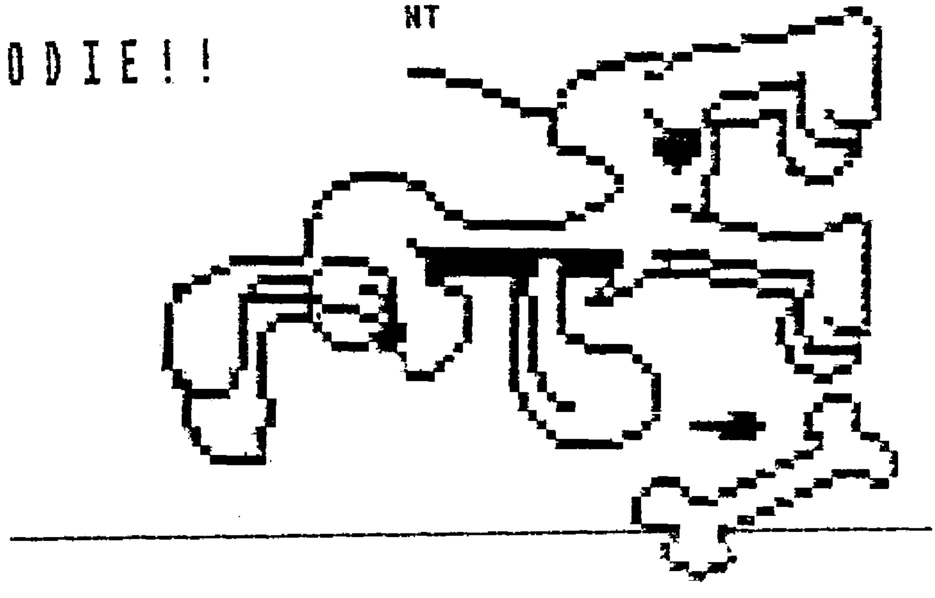
Jean H.

(Editor's note: I couldn't for the life of me remember who gave me the following article and listings - it pays to treat some editors as if they had a mental age of about 6 for the purposes of wording instructions and labelling everything..name address etc. I think Jean was the author...she did give me something!).

How do you show people who have never had contact with a computer what one can do? They are probably over 15, so a fast moving game zapping aliens does not impress, and they are likely to feel that in the real world computers must do something a bit more serious. You cannot show them anything too complicated, else they become even more afraid of these modern devils.

There is one topic to which the middle aged and elderly can relate - money. Here is a little program that calculates the return on investment, at compound rates of interest. It also gives a monthly savings scheme if you want to reach a target, say to buy a second processor for your Beeb.

If they are still with you after that, you might try the one to tell how many rolls of wallpaper are needed for a room.



```
   10 REM PROGRAM INTREST,DATE 19.10.84
   15  REM A program about compound inte
rest, and a monthly savings scheme
   20 MODE 7
   30 PROCinitial
  100 REPEAT
  110    PROCprintmenu
  120    IF choice$ = "1" THEN PROCamount
  130    IF choice$ = "2" THEN PROCprinc
  140    IF choice$ = "3" THEN PROCperiod
  150    IF choice$ = "4" THEN PROCrate
  155    IF choice$ = "5" THEN PROCsaving
s
  160    UNTIL choice$ = "6"
  165 CLS:PRINT:PRINT:PRINT:FOR I%=1 TO
2:PRINT TAB(5)CHR$131;CHR$141;"GOODBYE":
NEXT I%
  170 END
  171 :
  172 :
  200 DEFPROCinitial
  210 CLS:PRINT:FOR I=1 TO 2:PRINTTAB(5)
CHR$141;CHR$131;"INTEREST":NEXT I
  220 PRINTTAB(2,10)"This is a program t
o calculate"'"  how much an investment o
f $100"'"  would be worth at a given rat
e"'"  of interest after so many years"
  230 PRINT:PRINT"  It will also tell yo
u how much"'"  you have to invest to get
 $100"'"  after so many years"
  240 PROCreturn
  250 CLS:PRINTTAB(2,3)"You will be aske
d some questions"'"  about the interest
rate you expect"'"  or the number of yea
rs of the"'"  investment"
  265 PRINT:PRINT"  When you are asked f
or your initial"'"  investment, give a n
umber which "'"  is less than 100"
  270 PRINTTAB(1,12)"Answer each questio
n, then press RETURN"
  280 PROCreturn
  290 ENDPROC
  291 :
  292 :
  300 DEFPROCprintmenu
  310 CLS
  320 PRINT:PRINT"Choose one of the opti
ons below,":PRINT"by pressing the approp
riate key.":PRINT:PRINT"Then press RETUR
N":PRINT
  350 PRINT:PRINTCHR$129;"1.";CHR$135;"A
mount from investment of $100"
  352 PRINT:PRINTCHR$129;"2.";CHR$135;"H
ow much to invest to get $100"
  354 PRINT:PRINTCHR$129;"3.";CHR$135;"N
umber of years to $100"
  356 PRINT:PRINTCHR$129;"4.";CHR$135;"R
ate needed to get to $100"
  357 PRINT:PRINTCHR$129;"5.";CHR$135;"S
avings Scheme"
  358 PRINT:PRINTCHR$129;"6.";CHR$135;"E
nd the program"
  360 choice$ = FNinput("123456")
  370 ENDPROC
  375 :
  376 :
  400 DEFPROCamount
  410 CLS
  420 PRINT:PRINT:princ=100
  430 INPUT"What interest rate will you
get, %  "rate: r = rate/100
  440 PRINT:INPUT"How many years for the
 investment  "years
  450 amount = princ*(1+r)^years
  452 amount = INT(amount*100)/100
  460 PRINT:PRINT:PRINT"You would get";S
PC(5);CHR$130;"$"; amount;'SPC(9);"from
     $100":PRINT SPC(11);"at       ";CHR$1
34;rate;" %":PRINTSPC(11);"in       ";CHR$
133;years;" years"
  488 PROCreturn
  490 ENDPROC
  491 :
  492 :
  500 DEFPROCprinc
  510 CLS
  520 PRINT:PRINT:amount = 100
  530 INPUT"What interest rate will you
get, % "rate: r=rate/100
  535 PRINT
  540 INPUT"How many years for the inves
tment "years
  550 princ =amount/(1+r)^years
  552 princ = INT(princ*100)/100
  560 PRINT:PRINT:PRINT"You would need t
o invest    $";CHR$130;princ:PRINT SPC(1
8);"to get     $";amount:PRINT SPC(22);"
at     ";CHR$134;rate;"%":PRINT SPC(21);"
 in     ";CHR$133;years;" years"
```

```
  580 PROCreturn
  590 ENDPROC
  591 :
  592 :
  600 DEFPROCperiod
  610 CLS
  620 PRINT:PRINT:amount = 100
  630 INPUT"What interest rate will you
get,% "rate:r=rate/100
  635 PRINT
  640 INPUT"What is your initial investm
ent $"princ
  642 IF princ >99 THEN PRINT"Choose a n
umber less than 100":GOTO 640
  650 funca= LOG(amount)-LOG(princ)
  652 funcb=LOG(1+r)
  656 years= funca/funcb :years = INT(ye
ars*10)/10
  660 PRINT:PRINT:PRINT"It would take yo
u   ";CHR$133;years;" years":PRINT SPC(6
);"to get from   $";CHR$134;princ';SPC(1
5);"to   $";CHR$130;amount
  680 PROCreturn
  690 ENDPROC
  691 :
  692 :
  700 DEFPROCrate
  710 CLS
  720 PRINT:PRINT:amount=100
  730 INPUT"What is your initial investm
ent $"princ
  732 IF princ>99 THEN PRINT"Choose a nu
mber less than 100":GOTO 730
  735 PRINT
  740 INPUT"How many years "years
  750 funca = LOG(amount)-LOG(princ)
  760 funcb=funca/years:func=10^funcb
  770 rate=INT(1000*(func-1))/10
  775 PRINT
  780 PRINT:PRINT"The rate must be    "
;CHR$130;rate;"%":PRINT"to raise you fro
m    $";CHR$133;princ:PRINT SPC(15);"to
   $";CHR$129;amount:PRINT SPC(15);"in
   ";CHR$134;years;" years"
  790 PROCreturn
  795 ENDPROC
```

```
  796 :
  797 :
  800 DEF FNinput(i$)
  810 LOCAL an$,I%
  820 an$=""
  830 REPEAT
  840   I%=GET
  850   IF INSTR(i$,CHR$(I%)) THEN an$=
CHR$(I%)
  860   UNTIL I%=13 AND LEN(an$)=1
  870 =an$
  871 :
  872 :
  900 DEFPROCreturn
  910  REPEAT:PRINTTAB(2,22);CHR$(131);"
PRESS THE SPACE BAR TO CONTINUE"
  920     UNTIL GET = 32:ENDPROC
  921 :
  922 :
 1000 DEFPROCsavings
 1005 CLS:  PRINT:PRINT CHR$130;SPC(6);"
MONTHLY SAVINGS SCHEME"
 1010 PRINT:PRINT"This tells you how muc
h you have"'"to save each";CHR$129;"mont
h,";CHR$135;"to reach your target"'"in t
he time you have available."
 1020 PRINT:PRINT CHR$129;"What is your
target   ";CHR$135;" $";:INPUTamount
 1030 PRINT:PRINTCHR$129;"How many years
and months do you have":PRINT
 1040 INPUT "Years "years,"Months "mo
 1060 PRINT:INPUT"What interest rate can
you get %"rate
 1070 months = 12*years+mo
 1080 r = rate/1200
 1090 x= 1+r
 1100 func = x*(x^months -1)/r
 1120 saving = amount/func: saving = INT
(saving*100)/100
 1125 PRINT
 1130 PRINT:PRINT CHR$130;"You need to s
ave $";saving:PRINT CHR$133;"per month t
o reach your target"
 1140 PROCreturn
 1150 ENDPROC
```

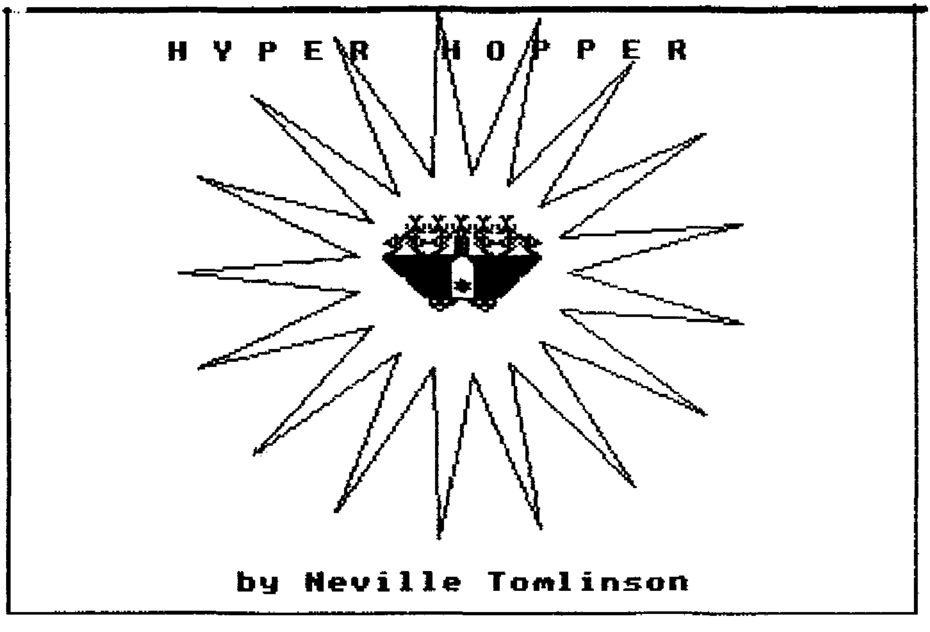
```
   10  REM Program WPAPER
   20  MODE 7:PRINT:PRINT
   30  CLS:FOR I = 1 TO 2:PRINTTAB(5)CHR
$141;CHR$133;"WALLPAPER":NEXT I
   40 PRINT:PRINT:PRINT"A PROGRAM TO CAL
CULATE ":PRINT"THE NUMBER OF ROLLS OF WA
LLPAPER"
   50 PRINT"NEEDED FOR A ROOM"
   60 PRINT:PRINT:PRINT"Give all answers
 in metres":PRINT"and 1 decimal of a met
re"
   90  PRINTTAB(2,20)CHR$131;"PRESS THE
SPACE BAR TO CONTINUE"
   95  REPEAT:UNTIL GET = 32
  195  REPEAT
  200    CLS:PRINT:PRINT:
  210    PRINTCHR$(130);"What is the tota
l length all around"
  220    PRINTCHR$(130);"the room, withou
t the windows."
  230    PRINT:PRINT CHR$(133);"Begin at
the edge of one window, and"
  240    PRINT CHR$(133);"measure right r
ound to the other edge"
  250    PRINT CHR$(133);"of the same win
dow,jumping"
  252    PRINT CHR$(133);"over all other
windows which do not"
  254    PRINT CHR$(133);"come down to th
e floor"
  260    INPUT total_length
  270    PRINT:PRINT CHR$(131);"What is t
he combined length of all":PRINT CHR$(13
1);"windows"
  280
  290    INPUT window_length
  300    PRINT:PRINT CHR$134"What is the
usual height of windows"
  310    PRINTCHR$134;"from the floor"
  320    INPUT window_height
  330    PRINT:PRINT CHR$(133);"What is t
he ceiling height":INPUT height
  340      FOR I=1 TO 2000:NEXT:CLS
  350
  400  drops_needed =INT(total_length/.5
34)
  410   drops_per_roll =INT(10/height)
  420 remnant=10-drops_per_roll*height
  430 rolls_used= INT(drops_needed/drops
_per_roll)+1
  435  drops_to_last =drops_per_roll*(ro
lls_used -1)
  440  extra = 10-(drops_needed -drops_t
o_last)*height
  445  IF extra = 10 THEN extra = remnan
t
  450 num_remnants=rolls_used -1
  460  PRINT:PRINT:PRINT CHR$129;"You wi
ll need ";rolls_used ;" rolls"
  470  PRINT CHR$129;"for the full lengt
h pieces"
  480   PRINT:PRINTCHR$129;"And there wi
ll be ";extra;" metres left,":PRINT CHR$
129;"plus scraps"
  490 w_drops_needed =(window_length/.53
4)
  500  IF window_height < remnant THEN P
ROCuse_scraps ELSE PROCno_scraps
  510     IF more = 0 THENPRINT:PRINT CHR
$129;"These will be enough for":PRINT CH
R$129;"the areas above and below the win
dows"
  520    IF more = 1 THEN PRINT:PRINT CHR
$129;"You will need one more roll":PRINT
 CHR$129"for above and below the windows
"

  530  PRINTTAB(2,20)CHR$131;"PRESS THE
SPACE BAR TO CONTINUE"
  540   REPEAT:UNTIL GET = 32
  550  UNTIL FALSE
  600  DEFPROCuse_scraps
  610   IF w_drops_needed < num_remnants
 THEN more =0 ELSE more =1
  620  ENDPROC
  700  DEFPROCno_scraps
  710  IFextra >(w_drops_needed*window_h
eight)THEN more = 0 ELSE more =1
  790  ENDPROC
```

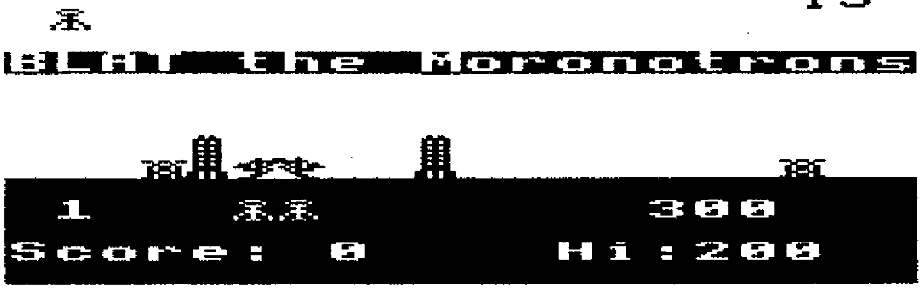
HYPER HOPPER

Neville Tomlinson has, as promised sent in a magnificant game for the Christmas issue. Trouble is, it is so long (fills up all available memory) that I doubt whether we have enough enthusiasts to justify the space it would take up in the mag. So what I propose is that we will supply it on tape to whoever wants it, at a nominal fee: either send a cassette and $1-00, a disc and $2-00, or just cash ($3-00 for cassette, $8-00 for disc).

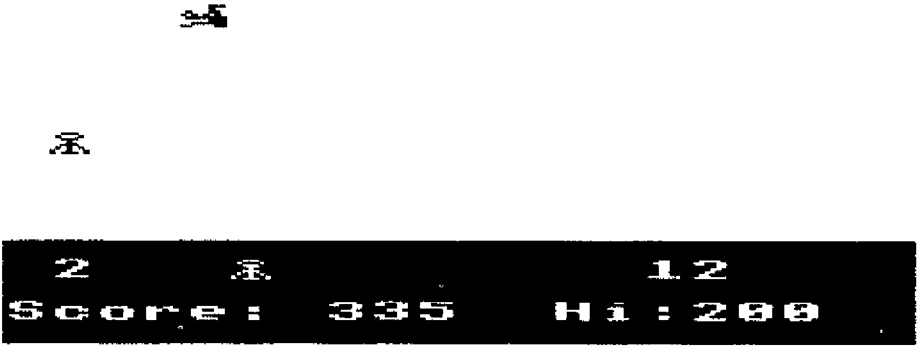
The game incorporates full MODE 2colour and graphics, and uses sound extensively as well. It incorporates a title page and a high score table.

Neville says "You, as Captain of the 'USS Hyper Hopper', have been given the mission of eradicating the Moronotron Aliens Base." You must survive four screens of nasties. First, you use your hopper to jump over enemy cities - its all over if you land on one! Next you must ram your hopper into flying defences before they move twice across the screen. As you hit each piece, a mean-looking dragon builds up in the top left corner of the screen. If you successfully demolish all the pieces, the FIRE breathing Dragon starts moving. If you don't manage to jump onto his back, dodging the fire, before he gets halfway down the screen he will pounce on you! Once on the Dragon's back you are carried to the Alien's Base, with which you must dock and then demolish with a bomb. Success brings a choice of one of four ships to start again : but this time things will get tougher!

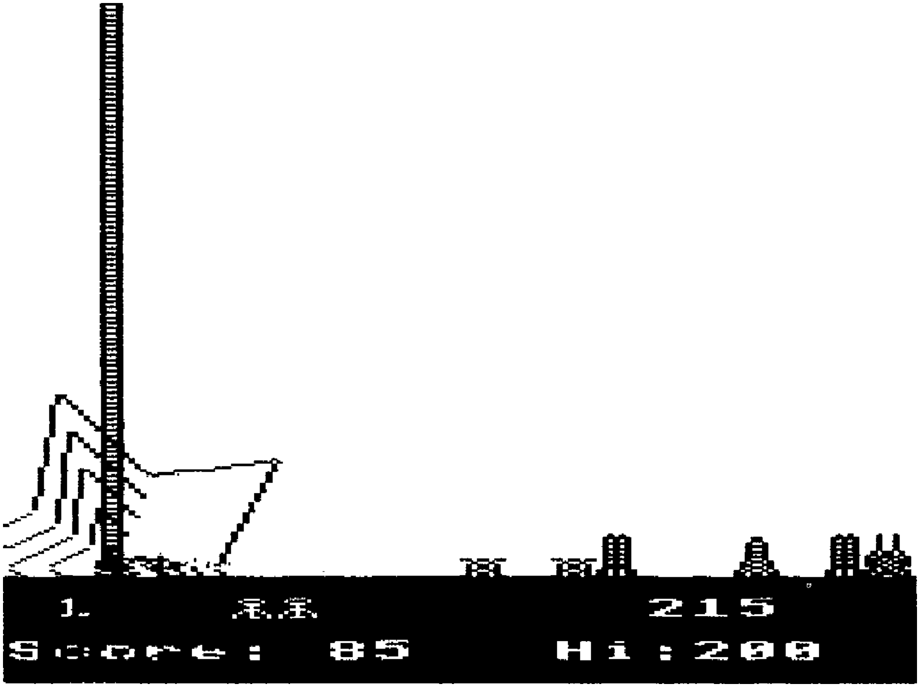




```
BLAT the Moronotrons

 1                300
Score:  0      Hi:200
```



```
 2                12
Score:  335    Hi:200
```



```
 1                215
Score:  85     Hi:200
```

THE 12 DAYS OF BEEBMAS

Tony W Napier

On the first day of Christmas, my true love sent to me:
A Beeb with a Three Month Guarantee.

On the second day of Christmas my true love sent to me:
Two Shugart drives and a Beeb with a Three Month Guarantee.

On the third day of Christmas my true love sent to me:
Three light pens,
Two Shugart drives and a Beeb with a Three Month Guarantee

On the fourth day of Christmas my true love sent to me:
Four CALL-ing words,
Three light pens,
etc...

....
....
....
On the twelfth Day of Christmas my True Love sent to me:

Twelve Programs running

Eleven printers printing,

Ten loops a-looping,

Nine Lists deleting,

Eight Modems muttering,

Seven Sprites a-sprinting,

Six spokesmen lying,

Five Gold ROMS!

Four CALL-ing words,

Three light pens,

Two Shugart drives,

And a BEEB with a Three Month Guarantee!

HOW TO GET RID OF A PAIN IN THE NECK

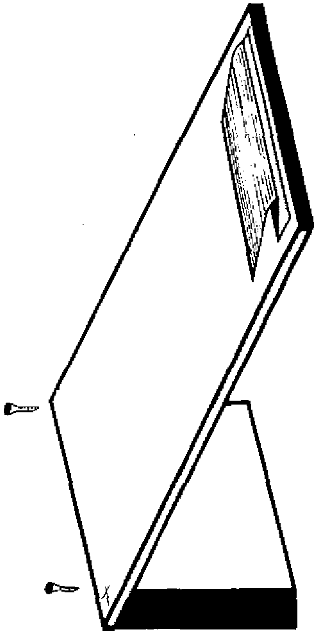
Geoff P.

Entering programs from books or magazines can be a pain in the neck. And the back and shoulders too. Sitting hunched over a keyboard copying line after line of yext, (sic) then swinging one's head up to stare at the telly or monitor to check you've got it right is an ergonomist's nightmare. After half an hour you get up to stretch and feel like you've been humping sacks of coal all day.

There is an easier way. I saw the answer in a commercial stationers. A device called a copy-holder - basically just an upright clipboard - guaranteed to leave you sitting more comfortably, not least because it lifted more than $30 worth of small change from your back pocket! Having nothing in my wallet but an impressive collection of moths, I determined there must be a cheaper way.
And there is: $3.35 and a half-hour work.

You will need:

1 Eastlight Clipboard	$3.25
2 15mm wood screws	.10
1 piece of wood 20mm thick 235x100mm	(Can be found anywhere. - a piece of your neighbour's house, for example!)



THE ECONOMY COPYBOARD

The diagram should be self explanatory, even to the most architecturally inept; the key to the whole device being the 75° angle on a piece of wood used as a base. Those like myself who are unable to saw straight should use a piece of coarse sandpaper to achieve the desired gradient.

The clipboard mentioned is the cheapest, most practical and most robust on the market. It has a plain hardboard base (ideal for drilling holes in), and a strong, wide clip that can hold the weight and depth of my 800-page dictionary with ease. Beware of perspex imitations at three times the price!

Simply drill a couple of holes, countersink them to keep things tidy, and insert the screws. Philistines may use nails, though I'd advise you to drill the hardboard first in case it splits.

What could be easier? Well I suppose you could always crib someone else's programs at the next User Group meeting.
Now there's an idea...

VAMPIRE ADVENTURE

reviewed in Hamilton by Andrew A.

From FG Software, Box 6662, Auckland price $24.95.

The object of this text-only adventure game is to fing and kill Count Dracula, but first you must outwit the hungry hounds, the evil bats and other unsavoury characters. Easier said than done! Although it isn't difficult to get started and find your way through many of the rooms, actually getting through to the end is very difficult. Moving from room to room in the castle, you must make the best of your 'senses' because after a number of moves the clock strikes twelve times and you fing yourself waking up in a strange room, without any of the objects you had before. A similar fate awaits you when the new day begins!

Vampire Adventure is supplied with a back-up copy on the reverse side of the tape, and has a SAVE facility. Although it lacks the large vocabulary of some adventures, Vampire Adventure is an amusing, well-crafted adventure that will test experienced adventurers while still being suitable for beginners.

ELECTRON-

A FASCINATING PIECE OF HIGH TECHNOLOGY

H. Wiggins' Adventures

I have had this new device, the Electron, for over 6 months now. At first I pined for the use of my printer, lying sadly idle. However, the Plus 1 interface arrived and I bought one of the first of these. This promised printer access, but alas, my printer has serial input. An article in July'83 "Electronics Australia" described an inexpensive parallel to serial converter which to my sorrow did not work until I discovered that a capacitor was needed in the acknowledge line. This is not mentioned in that article, nor in several I saw. Said printer now works beautifully. Everyone asks me how I like the Electron, and imply I should have bought a BBC. However, I reply that I bought the Electron for home use, being a retired person. The cost of a BBC at that time was not warranted at NZ prices. Comments are: keyboard good, BBC Basic good, machine code facilities excellent, lacks an RS432 cartridge which would have bypassed the effort described above. This too may arrive in time.

At first I complained about the lack of Electron programs and their price. Now there is plenty of exercise to be had in typing in the programs which are available in the literature for the Electron. Also, many BBC programs run well. There are some good, if expensive magazines. I subscribe to one, ELBUG, produced by Beebug. Here is a brief program from it to make your Electron tick like a clock:

```
10 REPEAT TIME=0:*M.1
20 REPEAT UNTIL TIME>100:TIME=0:*M.0
30 REPEAT UNTIL TIME>100:UNTIL 0
```

Do not leave this program running indefinitely - the relay might not like it.

I have had difficulty in running some of the more exotic BBC programs. One such is WORD PERFECT which goes nicely but refuses to print (one of the reasons this article was produced on a typewriter - one of those annoying machines which does not know how to spell). However I have consulted an Oracle as to why this one does not run. Perhaps TONY will be able to tell me.

The graphics appear to be the same as the BBC. The Electron has 20½K of space available for Basic programs. (cf BBC with 27½K in Mode 7 or 20½K in Mode 6). This has never worried me. It also doesn't have Teletext Mode 7 but since I do not know what it is, this has never worried me either. The 1.2 operating system and BASIC II can differ in odd respects from the older BBCs (0.1 OS and Basic I), and it runs a little slower - about 80 compared to 60 on a BBC in a test program. The sound has one channel with a very tiny speaker, but the BBC has a variety of noises.

I have also seen good X/Y drawings made on a plotter with an Electron. The plug-in cartridge facilities enable one to use LISP and FORTH as alternative languages. These same plug-in sockets show promise like the sideways RAMs do in the BBC. I have not so far used the 4 channels of A-D which are available. Altogether one can say that the Electron is a fascinating piece of high technology, and one which has the ability to develop further.

GORF

reviewed in Westport by Jamie D

Doctor Soft, $24.95. Review copy supplied by FG Software, Box 6662, Auckland.

The object of this game is to obliterate the 4 screens of attacking aliens. You have three lives to do it in.

The first screen is much like Space Invaders in that you have to shoot alien 'meanies' progressing slowly across the screen. Screen two is more difficult; you must shoot the aliens that swoop down on you. The third screen is the most difficult of all in my opinion. You must shoot the alien craft that circles around and around dropping fire balls (these cannot be shot). The fourth and final screen is where you do your utmost to destroy the Gorfian Mother Ship. Occasionally a bullet appeared to go through a meanie, though it may just have been my aiming.

Each time you complete a set of screens, you are presented with a harder (MUCH harder) set of screens to accomplish.

You control your ship with either joysticks or keyboard. Either way there is fast response. The sound and graphics are excellent, but the flashing on the first screen may be a little annoying.

A very enjoyable program which the whole family may like (mine did). The sound is a bit loud and may get your mother (or Wife) telling you to stop playing, but the game is so good you won't want to. A nice touch was the way the instructions scrolled slowly up the screen, giving you ample time to read them.

TOOLKIT - BEEBUG's

reviewer F.Hoonhout Hamilton

Many months ago I saw an advertisement in BEEBUG for a ROM-based ToolKit for BBC Basic. Delighted by the thought of this, I promptly invested around '17, sent off the draft and settled back for a long wait. Imagine my surprise when 16 days later a parcle arrived at my local PO, tax of 15% outstanding. I duly paid the minimal charge (around $4) and found one ToolKit chip and manual, plus fitting instructions.

Having fitted the ROM, on turning it on all to be had was a continuous whine from BEEB. Fault: 0.1 OS. A visit to the local computer shop soon put matters right (and my purse lighter) with a 1.2 OS. Typing *HELP proved the Beeb now accepted ToolKit, and out came the manual.

The following functions are available, some of which are explained further:

*Check
*Clear
*Edit
*Free
*Memory
*Merge
*Move
*New
*Off
*Old
*On (f)
*Pack
*Recover
*Renumber
*Report
*Screen
*Util (1-9)

All commands can be prefixed by the letter 'B' to avoid clashing with similar commands operating in other ROMs in your machine.

*clear clears and resets all variables, A%-Z% to zero.
*edit enters a remarkably fast and powerful screen editor.
*mem displays a Hex and ASCII memory dump, accessing all memory locations from &0 to &FFFF (handy for cheating at SPHINX!!)
*util enters the menu-driven utility editor. In it the following options are available:

1) string search
2) search and replace
3) move program lines
4) list procedures and functions
5) list variables A% - Z%
6) list other numeric variables
7) list string variables
8) list arrays
9) select sub-range for options 1 & 2
10) enter screen editor.

One hint that wasn't in the manual (and caused me to initially believe the chip was faulty): you need to run your program before options 6,7 8 produce results.

The string search and replace options have the useful ability to distinguish between BASIC Keywords and other words. Thus if you wanted to replace the word "print" embedded in PRINT statements by the word "written" you won't end up with "written" Keywords all through your file. The Keywords (to be recognised as such) need to be preceded by a ''

In my opinion the ROM-based ToolKit is the most useful piece of software for people who want fast and easy editing of their BASIC programs. It doesn't use up anr RAM, and when debugging programs, the screen editor is entered at the line where the error was detected. Having used ToolKit for several months now, I can't imagine ever doing without a "programmer's editor" again.

MODEL A's REVENGE

Are you the proud owner of a Mighty Model A (that rare and fast-disappearing species), and sick of smug looks on faces of Model B owners? Well here is your chance of a sneak revenge raid! Type in *FX254, press BREAK, and walk away quietly - you have now converted a Model B to a 16K machine! It can only regain its full 32K by switching the whole machine OFF then On again, or by typing in *FX255,x where x is... (I can't quite remember)

APPLIED ENGINEERING AND TECHNOLOGY LTD

3 HALLEYS LANE Phone (04)850-800
WELLINGTON

aptec

FOR BBC AND ALL TYPES OF MICROCOMPUTER SERVICING

ELITE

reviewer Keith F

This Acornsoft program is an interactive game of the highest merit, far removed from the two-dimensional non-stop action arcade games. Here we have a game that requires thought and decision making as well as quick reactions. The player plays the part of an interplanetary trader, and accordingly can choose destinations and cargo carried.

A wide range of goods is available to choose from, and prices vary from planet to planet. For example, computers are more expensive on an agricultural planet than an industrial one. The planets themselves, as well as being rich, poor etc, also have differing government styles ranging from corporate state (nice and safe) to anarchist (downright treacherous).

The hapless trader begins his task in a ship with a single laser facing ahead. This tends to be a drawback, as pirates usually attack from the rear! If the player survives, he presumably makes money and gradually improves his ship with extra and more powerful lasers, missiles, extra cargo space etc.

Travel is 3-dimensional (as is battle) and incorporates hyperspace travel, thus allowing a large number of galaxies to be explored. Space flight is "realistic" in that it is frequently boring. Beware he who falls asleep however, as death comes quickly to the tardy. Attacks by pirates are an unavoidable part of the game; the ferocity and number of attacks being relative to the government type of your destination.

Combat takes place in 3-D with lasers and missiles. Unlike most games, fire power is limited. Lasers overheat if fired for too long without pause; missiles have to be targeted, and only a maximum of 4 are permitted on any journey. Just to complicate matters further, attacks can come from more than one direction at a time, and with 4 screens to look in (one at a time) you can only aim and fire in one direction at a time. However you do have a 3-D radar permanently displayed at the bottom of the screen to give an all-round view. You only have one life per game, so it pays to become a skillful pilot and an economic marksman. It also pays to Save the game before leaving the safety of a space station. If you get killed, you are automatically re-started at that position - without having to re-load. Couple these with astute business skills and you may reach "Average" after 18 hours play! (Broken, not all in one session). There are still five more grades to reach "Elite". You initially start the game classified as "Harmless"

The variety of weapons and ships available depends on whether you get the disc or the tape version. (The disc is more elaborate). Play can take place on the keyboard, or with the aid of a proportional joystick. An excellent set of instructions/manual, a short story showing what you can do, are supplied as well as a functionkey strip and reference card. The prudent will study them well.

This is not a complete description by any means; there is not the room to go into the scanners, system descriptions, status reports, police, contraband, space stations, energy bombs, control panels, technology levels, planet populations, cargo scoops, suns, 'jumping', bounty hunters, anti-missile defences, meteorites, shields, energy banks, docking computers, alien items........

Suffice it to say, Elite is the best computer or video game I have ever played (and I've played a fair few in my time!)

§§

FASTER A-D CONVERSION

Use *FX190,8 to force 8-bit conversion instead of 12-bit. This gives faster conversion over the same range as before, but with less precision. Reset with *FX190,0

KKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKK

ELECTRON *FX EXTRA

The Electron user Guide lists only a small number of the *FX calls that it can use. Apparently most of the *FX (OSBYTE) calls for the Beeb will also work on the Elk, unless they refer to features not present in the Elk (eg RS423 - but if you have the Plus 1 and then buy the RS423 extra, such a call will work!)

ΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩΩ

DISABLED ESCAPE

This can be done by using *FX200,1 and reset by *FX200,0

1985 CALENDAR

Neil B Akld

This calendar fits nicely on a sheet of A4 paper. It was written for Beeb plus the Epson-alike Logitec. The paper must be inserted twice through the printer. On the first pass January to June are printed; on the second, July to December. After the first pass, replace the paper and press the Select button on your printer to continue the print run. (Ed's "typist's" note: changes appended for CITOHs include reverse line feed to avoid this!)

Printer control codes have been REMmed for conversion help. I would advise omitting line 150 until the program is fully and correctly typed in, or got to a stage where printing out is feasible. You will notice that Good Friday is added at the bottom - I eliminated PROCeaster for brevity, but this can easily be added by readers wanting it, by entering the relevant code from Anton's program in July'84 Beeblet.

CITOH alterations:

```
160 REM
170 VDU2,1,27,1,69:REM elite
200 VDU1,15:REM elongated released
270 IF X=7 THEN VDU1,12,1,27,1,114,1,12,1,27,1,102:REM f
orm feed, reverse line feed, form feed, forward line feed
s again
```

```
   10 REM ** CALENDAR  by Neil B 11'84
   20 MODE0
   30 @%=6
   40 DIM M$(12),N%(12),DAY$(7)
   50 FOR A=1 TO 12
   60   READ M$(A),N%(A)
   70   NEXT
   80 FOR A=1 TO 7
   90   READ DAY$(A)
  100   NEXT A
  110 REPEAT
  120   INPUT''"Enter year ",Y%
  130   UNTIL Y%>1751
  140 D=FNfirstday(Y%)
  150 VDU2
  160 VDU1,27,1,64:REM  reset the pr
inter
  170 VDU1,27,1,80,1,0:REM ELITE
  180 VDU1,14:REM elongated
  190 PRINTTAB(14)"CALENDAR FOR ";Y%
  200 VDU1,29:REM  elongated released
  210 IF Y% MOD 4 = 0 THEN N%(2)=29
  220 ML%=5
  230
  240 FOR X=1 TO 12
  250   WL%=0
  260   IF X<7 THEN T%=0 ELSE T%=46
  270   IF X=7 THEN VDU1,12:REM form f
eed
  280   L%=LEN(M$(X)) DIV 2
  290   PRINT'TAB(T%+22-L%)M$(X)'
  300   PRINTSPC(4+T%);
  310   FOR B=1 TO 7
  320     PRINTDAY$(B);SPC4;
  330     NEXT
  340   PRINT
  350   K=0
  360   PRINTTAB(T%);
  370   REPEAT
  380     K=K+1
  390     IF K<D PRINTSPC6;
  400     UNTIL K=D
  410   L=0
  420   REPEAT
  430     L=L+1
  440     PRINT L;
  450     IF (L+D-1) MOD 7 = 0 THEN PR
INT'TAB(T%);:WL%=WL%+1
  460     UNTIL L=N%(X)
  470   D=D+(N%(X) MOD 7)
  480   IF D>7 THEN D=D-7
  490   FOR A%=WL%TO ML%
  500     PRINT
  510     NEXT A%
  520   NEXT
  530 VDU1,12,3
  540 @%=10
  550 END
  560
  570 DEF FNfirstday(F%)
  580 L=F%-1
  590 P=L/100
  600 Z1%=P/4
  610 Z2%=P
  620 Z3%=5*L/4
  630 Z=36+Z3%-Z2%+Z1%
  640 =Z-(7*INT(Z/7))+1
  650
  660 DATA JANUARY,31,FEBRUARY,28,MARC
H,31
  670 DATA APRIL,30,MAY,31,JUNE,30
  680 DATA JULY,31,AUGUST,31,SEPTEMBER
,30
  690 DATA OCTOBER,31,NOVEMBER,30,DECE
MBER,31
  700 DATA SU,MO,TU,WE,TH,FR,SA
```

STRING PUZZLE

```
   10 REM String Puzzle
   20 A$ = "A" + CHR$(&0D) + "Z" + CHR$(&0D)
   30 PRINT A$
   40 PRINT LEN(A$)
   50 IF A$ = "Z" THEN PRINT "Same" ELSE PRINT
"Different"
   60 END
```

What is happening at line 20? (remember that CHR$(&0D) is a carriage return). What will appear on the screen; what will appear on a printer if one is enabled before Running the program? Yes! You do get different results! Line 50 won't make much sense until you do run it.

Those of you who are familiar with Indirection Operators might like to define a byte array using T% to store the start address. Then replacing ALL occurrences of A$ with $T%. You may well be surprised at the results - is LENgth still the same?

TEXT DISPLAY SPEED

GCF

It should be obvious (ie it took me 18 months!) that some modes display text faster than others. In fact Mode 7 is fastest on the Beeb, and Mode 6 on the Elk (Electron). However the idea for actually finding the differences did not occur to me until after buying a 6502 co-processor.

The program here prints "X" 100,000 times on the screen in each Mode. It was run both with and without an active co-processor. If you don't have a printer, delete the VDU2. Elk-owners could also run the program, though you may prefer not to try to time Mode 7! Anybody who wants to rerun it with variations may want to reduce N% to a smaller value (eg 1E3: N%=1000), as the current value of 1E5 takes an age and a day to run. (Actually about 15 minutes on a Beeb, and perhaps nearer 25 minutes on an Elk)

```
   10 REM  This shows that it takes longer to print in
some modes than others!
   20 DIM mode_time%(7)
   30 N%= 1E5
   40 FOR mode% = 0 TO 7
   50   MODE mode%
   60   TIME = 0
   70   FOR I% = 1 TO N%
   80     PRINT"X";
   90     NEXT
  100   mode_time%(mode%)=TIME
  110   NEXT mode%
  120 MODE6
  130 VDU2
  140 PRINT"MODE seconds  -  TO PRINT ";N%;" X's"
  150 FOR mode% = 0 TO 7
  160   PRINTTAB(2);mode%;TAB(6);mode_time%(mode%)/100
  170   NEXT mode%
  180 VDU3
  190 END
```

Number of seconds to print 100,000 "X"'s

MODE	BEEB	BEEB+co	ELK
0	96.97	40.52	?
1	124.56	66.42	?
2	193.52	135.38	?
3	97.01	40.52	?
4	97.55	40.52	?
5	125.62	67.49	?
6	97.56	40.52	?
7	81.27	40.52	!

ANALYSIS
For Modes 0 to 6 on the Beeb, the time is largely dependant on the number of bits per pixel, and hence the work required to set up a character matrix of 8x8 pixels in the video RAM. Hence the time required for printing in any mode with one bit per pixel (2-colour mode) is almost exactly the same. The small differences can be attributed to extra "carriage returns" and scrolling required in different modes. Elk owners should find a more dramatic difference as the effective speed of the machine depends on the amount of video RAM being displayed. (The clock rate is 1 or 2 MHz, dependant on Mode, and for the 20 & 16K modes the 6502 processor is actually disabled for part of each scan line displayed on your tv).

The main reason a dual processor Beeb is faster at printing is that while the I/O processor is working out what bits to set in video RAM, the Language processor is busy working out what to do next. They work in parallel. Astute readers will note that a dual Beeb is just as fast in Mode 6 as in Mode 7, 0, 3, and 4 - this must mean that the superfast 3MHz 6502 is actually holding up the I/O processor, when printing in a 2-colour mode!!

* * *

SYSTEM INTEGER DISPLAY

GCF

```
10 REM Display all system integer variables
20 MODE4
30 FOR V = ASC("@") TO ASC("Z")
40   A$ = CHR$(V) + "%"
50   PRINT A$, EVAL(A$)," &";~EVAL(A$)
60   NEXT
70 END
```

A quick and nasty program to display the contents of all the resident system integer variables, both in decimal and hexadecimal. The program itself does not use any system integer variables (A%, B%, C%,...)

LEFT RIGHT FROGS!

Anne F.

Children often take a while to sort out Left from Right consistently. This program offers some practice at getting them straight. There are two ponds, in the shape of hands: the left hand on the left side of the screen, the right hand on the right. A frog hops from between them in and out of the ponds, and eventually doesn't come out from one of them. The child has to say which pond it is in - left or right. The pond he indicates empties, to reveal... a frog if he's right, or nothing if wrong. If he is wrong, the correct pond empties next to reveal the frog. The program tries to make the correlation between the words Left and Right, the sides (of the keyboard), and the hand shape.

There is a mild problem in this - "L" for Left is on the right-hand side of the keyboard, and "R" for right is on the left! A solution to this was triggered by an Acorn User program: use an overlay on top of the keyboard. The child then has only to press a key in the Left or Right half of the keyboard to indicate which side he thinks the frog is in.

THE OVERLAY

is just a sheet of paper (a bit bigger than A4 preferably), slid under the function-key strip to hold the top edge down, and taped down over the front edge of the Beeb. An example of what it could look like is at the end of this article. The hands aren't essential. The keys f4,6,Y,G,B will not respond to being pressed - they are a nominal dividing line. Space Bar doesn't work either - how can the computer tell whether the child pressed the left side of it or the right?

THE GRAPHICS

make use of scaling (remember the angels last December issue?), resetting graphics origins to position them, and OS1.2 command PLOT 77 to fill and empty the ponds. There is only one hand defined; the program reverses the shape for each drawing and adjusts the size for the program closure. Likewise there is only one PROCmove_frog: parameters to it indicate where it is starting from and going to.

MAKE IT HARDER

by removing from PROCsplosh the lines PROCdroplet, wait, PROCdroplet. These lines make some "splashes" appear when the frog lands in the pond, then disappear, and provide a reminder and a prompt of which side the frog is in.

SCORING

As written, there is a good_score% and a bad_score%, which are incremented by one each time depending on whether the answer was correct or not. At the end of the program if the child got more correct than wrong, he is congratulated, otherwise offered a (hopefully) encouraging word. I suspect that good_score% should be perhaps twice bad_score% (ie 67% success rate) before being told he's doing well.

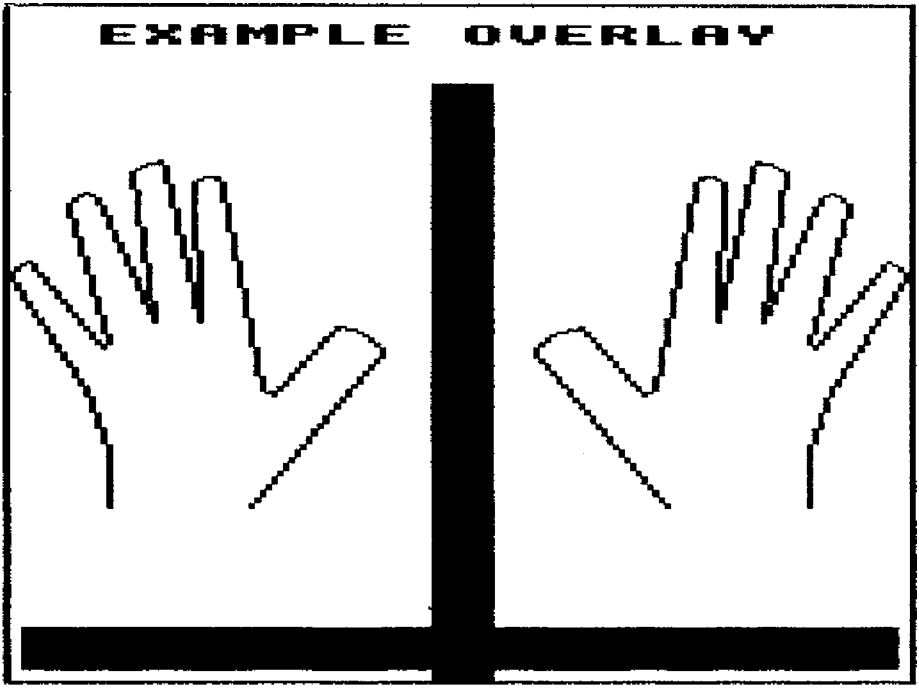
NOBBLED KEYS

lefthand$ is all those keys on the keyboard which indicate a left-hand answer. unwanted$ is the dividing line. This means all other keys default to righthand$.
The function keys have been redefined to return a number so they can be included (*FX225,128), the cursor keys have been set to return a value (*FX4,1) as has ESCape key (*FX229,1).

I recommend REMming these lines of the program until you have got it typed in accurately. FNquit and PROCtidyup allow the child to exit the program and reset all the keys back to normal behaviour.

WARNING?

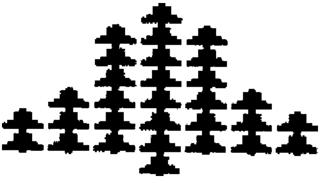
You need something better than OS 0.1 for this. Disc owners also need to reset page lower than &1900 eg PAGE=&1200 before running the program.



```
   10 ON ERROR PROCtidyup:CLS:REPORT:P
RINT" @ line ";ERL:STOP
   20 REM  Frog in a pond
   30 REM  Left-Right decisions
   40 REM  by Anne F
   50 REM  latest version 30/11/84
   60
   70 REM ****   disc owners! reset PA
GE or get No Room message soon
   80
   90 PROCsetup
  100 MODE1
  110 PROCscreen
  120 wait=INKEY(3000)
  130 MODE2
  140 VDU19,1,2;0;19,2,4;0;19,3,2;0;19
,4,4;0;19,5,4;0;19,6,2;0;19,8,6;0;19,1
0,6;0;19,12,4;0;
  150 SX%=8:SY%=8
  160 VDU23;8202;0;0;
  170 REPEAT
  180 CLS:CLG
  190 PROChand(12,0,1279,64)
  200 PROCfillup
  210 PROCjump
  220 PROCtest
  230 UNTIL FNquit
  240 PROCtidyup
  250 MODE1
  260 PROCscreen
  270 END
  280
  290 DEF PROCsetup
  300 good_score%=0
  310 bad_score%=0
  320 intro%=TRUE
  330 DIM Xfill%(11), Ystart%(11), Yfi
n%(11)
  340 RESTORE 2600
  350 FOR I%=1 TO 11
  360 READ Xfill%(I%), Ystart%(I%), Yf
in%(I%)
  370 NEXT I%
  380 *FX225,128
  390 *KEY10O. MRUN M
  400 *FX4,1
  410 *FX229,1
  420 lefthand$="Q1W2E3R4T5ASDFZXCV"+C
HR$27+CHR$9+CHR$128+CHR$129+CHR$130+CH
R$131
  430 unwanted$="6YGB"+CHR$132+CHR$32
  440 VDU23,244,&99,&BD,&FF,&3C,&3C,&9
9,&5A,&24
  450 VDU23,250,&3C,&42,&5A,&BD,&BD,&5
A,&42,&3C
  460 ENVELOPE1,4,45,0,0,6,0,0,1,0,-30
,0,0,120
  470 ENVELOPE2,10,0,0,0,0,0,0,34,10,-
20,-26,100,126
  480 ENDPROC
  490
  500 DEF PROCscreen
  510 SX%=2:SY%=2
  520 PRINT'''''''
  530 IF intro% THEN PROCstart
  540 IF (NOT intro% AND good_score% >
 bad_score%) THEN PROCcongrats
  550 IF (NOT intro% AND good_score% <
= bad_score%) THEN PROCnext_time
  560 PROChand(2,100,1180,768)
  570 PROChand(2,100,1180,368)
  580 intro%=FALSE
  590 PRINT''''
  600 ENDPROC
  610
  620 DEF PROChand(C%,A%,B%,D%)
  630 GCOL0,C%
  640 FOR I% = SX% TO -SX% STEP -2*SX%
  650 RESTORE 2570
  660 IF I%>0 THEN VDU29,A%;D%; ELSE V
DU29,B%;D%;
  670 READ X%,Y%
  680 MOVE X%*I%,Y%*SY%
  690 FOR J%= 1 TO 30
  700 READ X%,Y%
  710 DRAW X%*I%, Y%*SY%
  720 NEXT J%
  730 NEXT I%
  740 ENDPROC
  750
  760 DEF PROCfillup
  770 LOCAL X%,Y%,Z%
  780 FOR I% = SX% TO -SX% STEP -2*SX%
  790 IF I%>0 THEN VDU29,0;64; ELSE VD
U29,1279;64;
  800 FOR J%=1 TO 11
  810 IF J%=1 THEN GCOL0,4 ELSE GCOL0,
2
  820 X% = Xfill%(J%)*I%
  830 Y% = Ystart%(J%)*SY%
  840 Z% = Yfin%(J%)*SY%
  850 FOR K% = Y% TO Z% STEP 4
  860 PLOT 77, X%, K%
  870 NEXT K%
  880 NEXT J%
  890 NEXT I%
  900 GCOL0,7
  910 VDU5:MOVE-1152,-36:PRINT"Left":M
OVE-400,-36:PRINT"Right":VDU4
  920 ENDPROC
  930
  940 DEF PROCjump
  950 GCOL3,1
  960 VDU29,640;100;
  970 VDU5:MOVE 0,0:PRINTCHR$244;:VDU4
  980 jumper% = RND(4)+1
  990 FOR jumps% = 2 TO jumper%
 1000 frog_side% = RND(2)
 1010 IF frog_side% = 1 THEN PROCmove_
frog(160,640,100) ELSE PROCmove_frog(1
056,640,100)
 1020 PROCsplosh
 1030 IF jumps% < jumper% AND frog_sid
e% = 1 THEN PROCmove_frog(640,160,100)
 1040 IF jumps% < jumper% AND frog_sid
e% = 2 THEN PROCmove_frog(640,1056,100
)
 1050 NEXT jumps%
 1060 ENDPROC
 1070
 1080 DEF PROCmove_frog(x1%,x2%,Y%)
 1090 LOCAL I%,K,R%,X%,H%,OX%,OG%,G%
 1100 H%=600
 1110 GCOL3,1
 1120 R%=(x1%-x2%)/2
 1130 VDU29,(x1%+x2%)/2;Y%;
 1140 VDU5
 1150 K=H%/(R%*R%)
 1160 OX%=-R%:OG%=0
 1170 S%=8*SGN(R%)
 1180 FOR X%=-R% TO R% STEP S%
 1190 MOVE OX%,OG%
 1200 *FX19
 1210 PRINTCHR$244
 1220 G%=H%-K*X%*X%
```

```
 1230 MOVE X%,G%
 1240 OX%=X%:OG%=G%
 1250 PRINTCHR$244
 1260 NEXT X%
 1270 VDU4
 1280 ENDPROC
 1290
 1300 DEF PROCtest
 1310 IF frog_side% = 1 THEN side$ = "
left" ELSE side$ = "right"
 1320 VDU28,0,8,39,0:REM  set up text
window
 1330 COLOUR1
 1340 PRINT'" Where is the frog?"
 1350 PRINT'"  Left?";SPC5;"Right?"
 1360 PROCprompt
 1370 IF INSTR(lefthand$,reply$) THEN
reply_side% = 1 ELSE reply_side% = 2
 1380 PROCempty(reply_side%)
 1390 IF reply_side% = frog_side% THEN
 PROCcorrect ELSE PROCwrong
 1400 ENDPROC
 1410
 1420 DEF PROCcorrect
 1430 good_score%=good_score%+1
 1440 VDU28,0,8,19,0:CLS
 1450 COLOUR6
 1460 PRINT" Yes,"''"the frog was in t
he"'
 1470 IF side$="right" THEN PRINTSPC9;
 1480 COLOUR8:PRINTside$;:COLOUR6:PRIN
T" pond"
 1490 COLOUR14
 1500 PRINT'''SPC8"YES!";
 1510 wait=INKEY(300)
 1520 ENDPROC
 1530
 1540 DEF PROCwrong
 1550 bad_score%=bad_score%+1
 1560 VDU28,0,8,19,0:CLS
 1570 SOUND0,2,5,10
 1580 wait=INKEY(100)
 1590 COLOUR6
 1600 PRINT"No,"''"the frog was in the
"'
 1610 IF side$="right" THEN PRINTSPC9;
 1620 COLOUR8:PRINTside$;:COLOUR6:PRIN
T" pond"
 1630 PROCempty(frog_side%)
 1640 ENDPROC
 1650
 1660 DEF PROCempty(side%)
 1670 GCOL0,0
 1680 IF side% = 1 THEN VDU29,0;64;:I%
=SX%:Xside% = 152:inc%=64
 1690 IF side% = 2 THEN VDU29,1279;64;
:I%=-SX%:Xside% = -152:inc%=-64
 1700 FOR J%= 11 TO 1 STEP -1
 1710 IF J% > 1 THEN GCOL0,130
 1720 X% = Xfill%(J%)*I%
 1730 Y% = Ystart%(J%)*SY%
 1740 Z% = Yfin%(J%)*SY%
 1750 FOR K% = Z% TO Y% STEP -4
 1760 IF J% = 1 THEN GCOL3,4:PROCglug:
GCOL0,132
 1770 PLOT 77,X%,K%
 1780 NEXT K%
 1790 NEXT J%
 1800 GCOL0,2:GCOL0,128
 1810 REM flush that dratted channel 0
 and 1 sound
 1820 *FX21,4
 1830 *FX21,5
 1840 ENDPROC
 1850
 1860 DEF PROCglug
 1870 MOVE Xside%,K%:DRAW Xside%+inc%,
K%
 1880 SOUND0,-15,3,1:SOUND1,-1,100+K%,
1
 1890 ENDPROC
 1900
 1910 DEF PROCsplosh
 1920 SOUND0,-15,100,10:SOUND1,-10,80,
10
 1930 random_seed% = ABS(RND)
 1940 PROCdroplets
 1950 wait=INKEY(50)
 1960 PROCdroplets
 1970 ENDPROC
 1980
 1990 DEF PROCdroplets:LOCAL X%,Y%,I%
 2000 dummy% = RND(-random_seed%)
 2010 GCOL3,10:VDU5
 2020 FOR I% = 1 TO 9
 2030 IF frog_side% = 1 THEN X%=RND(40
0)-300 ELSE X% = RND(400)
 2040 Y%=RND(500)+100:MOVEX%,Y%:PRINTC
HR$250;
 2050 NEXT I%
 2060 VDU4
 2070 ENDPROC
 2080
 2090 DEF PROCtidyup
 2100 VDU4,26,20
 2110 *fx225,1
 2120 *fx4
 2130 *FX229,0
 2140 *FX15,1
 2150 ENDPROC
 2160
 2170 DEF FNquit
 2180 VDU26
 2190 COLOUR7
 2200 PRINTTAB(0,30)"Left to end  OR"'
"  right to continue";
 2210 PROCprompt
 2220 IF INSTR(lefthand$,reply$) THEN
= TRUE ELSE = FALSE
 2230
 2240 DEF PROCprompt
 2250 *FX15,1
 2260 REPEAT
 2270 *FX202,32
 2280 reply$ = GET$
 2290 UNTIL INSTR(unwanted$,reply$) =
0
 2300 ENDPROC
 2310
 2320 DEF PROCcongrats
 2330 COLOUR3
 2340 PRINTSPC12;"CONGRATULATIONS,"
 2350 PRINT''SPC8"You do know the diff
erence"
 2360 PRINT''SPC16"between"
 2370 PRINT''SPC7"left";SPC7"and";SPC7
"right"
 2380 ENDPROC
 2390
 2400 DEF PROCnext_time
 2410 COLOUR3
 2420 PRINTSPC9"left";SPC5"and";SPC5"r
ight"
 2430 PRINT''SPC18"are"
 2440 PRINT''SPC15"difficult,"
 2450 PRINT''SPC10"better luck next ti
me"
```

```
 2460 ENDPROC
 2470
 2480 DEF PROCstart
 2490 PRINT"   Fred Frog hops from pon
d to pond."
 2500 PRINT''SPC14"When he hides"
 2510 PRINT''"  press a Key on the sid
e he is hidden"
 2520 PRINT''SPC7"Left";SPC8"or";SPC7"
Right"
 2530 PRINT''''''''"Press any Key to st
art"
 2540 ENDPROC
 2550
 2560 REM hand outlines
 2570 DATA 18,0,18,4,18,10,14,22,1,43,
2,45,4,46,5,45,17,30,18,31,11,56,12,58
,14,59,16,58,26,35,22,63,24,64,27,65
 2580 DATA 28,63,34,35,33,61,35,62,37,
62,38,60,45,22,47,21,58,34,62,33,65,31
,66,29,43,0
 2590 REM hand & thumb
 2600 DATA 31,0,8,30,8,21,58,21,34,31,
21,35
 2610 REM little  & fourth fingers
 2620 DATA 10,30,38,5,38,44,3,44,45,20
,35,47,14,47,59
 2630 REM middle  & index fingers
 2640 DATA 27,35,65,35,35,62
```



A ROSE BY ANY OTHER NAME

Some things can be done in a variety of different ways to achieve the same effect:

```
PRINT"A"
PRINTCHR$65                for instance, these three
VDU65

PRINT"     "
PRINTSPC5                  these four do the same
PRINTSTRING$(5," ")
VDU32,32,32,32,32

PRINTTAB(X,Y)"A"
VDU31,X,Y,65
```

ZAPPER

Neville T.   Ch'ch

For those of you who like little puzzles, here is one with graphic representation. Full instructions are included in the program.

Once you have discovered valid sequences, you can make things harder and fun again by starting with random squares already filled. To do this, add the following line:

```
305 FOR K=1 TO 5:PROCfill(RND(9)):NEXT
```

```
   10 REM *** ZAPPER! ***
   20 REM by Neville Tomlinson
   30 REM Sept '84
   40 REM version 0.2
   50
   60 *TV255
   70 DIM number%(9),cnrx%(9),cnry%(9)
   80 MODE7
   90 PROCheader:PROCinstr
  100 REPEAT
  110   MODE1:VDU23;2,92;0;0;23;8202;0;0;0;
  120   PROCinit
  130   REPEAT
  140     PROCkeys
  150     PROCplot
  160     UNTIL check%
  170   wait=INKEY(100)
  180   MODE7:PROCheader:PROCagain
  190   UNTIL NOT yes%
  200 END
  210
  220 DEF PROCinit
  230 RESTORE 330:VDU5,19,3,4;0;
  240 moves%=0:correct%=FALSE
  250 FOR K=1 TO 9
  260   number%(K)=0:READ cnrx%(K),cnry%(K)
  270   GCOL0,1:MOVEcnrx%(K),cnry%(K):DRAWcnrx%(K)+250,cnry%(K):DRAWcnrx%(K)+250
,cnry%(K)-250:DRAWcnrx%(K),cnry%(K)-250:DRAWcnrx%(K),cnry%(K)
  280   GCOL0,3:MOVEcnrx%(K)-171,cnry%(K)-109:PRINTK
  290   NEXT K
  300 VDU4
  310 ENDPROC
  320
  330 DATA 250,1000,510,1000,770,1000
  340 DATA 250,740,510,740,770,740
  350 DATA 250,480,510,480,770,480
  360
  370 DEF PROCkeys
  380 COLOUR130:COLOUR3
  390 PRINTTAB(12,30)"Which Square ?";
  400 REPEAT key% =GET-48
  410   UNTIL key%>0 AND key%<10
  420 moves%=moves%+1
  430 ENDPROC
  440
  450 DEF PROCplot
  460 PROCfill(key%)
  470 IF key%=1 THEN PROCfill(2):PROCfill(4)
  480 IF key%=3 THEN PROCfill(2):PROCfill(6)
  490 IF key%=7 THEN PROCfill(4):PROCfill(8)
  500 IF key%=9 THEN PROCfill(6):PROCfill(8)
  510 IF key%=2 OR key%=8 THEN PROCfill(key%-1):PROCfill(key%+1)
  520 IF key%=4 OR key%=6 THEN PROCfill(key%-3):PROCfill(key%+3)
  530 IF key%=5 THEN FOR K=2 TO 8 STEP2:PROCfill(K):NEXT
  540 check%=TRUE:FOR K=1 TO 9:check%=check% AND number%(K):NEXT
  550 ENDPROC
  560
  570 DEF PROCfill(num%):number%(num%)=NOT number%(num%)
  580 GCOL3,2
  590 MOVE cnrx%(num%),cnry%(num%)
  600 MOVEcnrx%(num%)+250,cnry%(num%):PLOT85,cnrx%(num%)+250,cnry%(num
%)-250:MOVEcnrx%(num%),cnry%(num%)-250:PLOT85,cnrx%(num%),cnry%(num%)
  610 DRAW cnrx%(num%)+250,cnry%(num%)-250
  620 ENDPROC
  630
  640 DEF PROCagain
  650 PRINT''':FOR K=1 TO 2:PRINTCHR$141CHR$131"You completed the
ZAPPER in only":NEXT
  660 PRINT
  670 FOR K=1 TO 2:PRINTCHR$141CHR$130TAB(15);moves%;" moves!":NEXT K
  680 PRINT'''':FOR K=1 TO 2:PRINTCHR$141CHR$133"Another game  Y/N
?":NEXT
  690 REPEAT: K=GET:UNTIL K=89 OR K=78
  700 IF K=89 THEN yes%=TRUE ELSE yes%=FALSE
  710 ENDPROC
  720
  730 DEF PROCheader
  740 FOR K=1 TO 2:PRINTCHR$141CHR$134CHR$157CHR$132SPC12"Z A P P E
R":NEXT
  750 PRINT'CHR$129TAB(12)"by Neville Tomlinson"
  760 ENDPROC
  770
  780 DEF PROCinstr
  790   PRINT''CHR$130"The object of the game is to fill
```

```
all"'CHR$130"nine squares."
  800 PRINT'CHR$134"Your controls are the Keys 1 to 9."
  810   PRINT''CHR$133"However,   the   tricky   part   is   that
when"'CHR$133"you turn any of the nine squares on, it"CHR$133"reverses
the squares directly adjoining"CHR$133"it!!"
  820   PRINT''CHR$131"It's   not   as   easy   as   it   sounds   -
the"'CHR$131"minimum number of moves is 5...."
  830 PRINT'''CHR$131CHR$157CHR$129"Press the space bar to start...
"CHR$156
  840 REPEAT UNTIL GET=32
  850 ENDPROC
```

THE ENIGMA SOLUTION

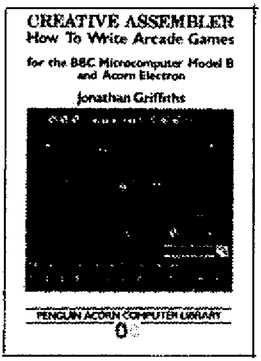
Some of you will recall the ENIGMA puzzle, derived from "New Scientist", which I posed in the October Beeblet. To recap, ENIGMA is exactly divisable by EE NN II GG MM AA BB and CC. Each individual letter represents a different digit (0 to 9). Which number does ENIGMA represent? There is only one correct solution.

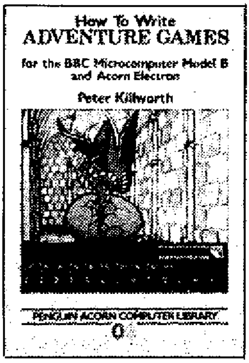
A little thought simplifies the problem.1. It is not meaningful to divide by zero therefore no letters represent zero.
2. ENIGMA must be even (it must be divisable my 2 or 4).
3. It cannot end in 0 or 5 therefore is not divisable by 5. No letter represents five.
4. The biggest number ENIGMA can represent is 987643, the smallest 123467.
5. The lowest common denominator of the remaining digits (ie 0-9 less 0,5) is 5544. ENIGMA must be a multiple of this.

A number of other deductions are also possible (eg. either B or C is 1), however this is a good a point as any to launch into a program. Here is the program suggested by Steve Wozniak, inventer and marketer of the Apple (he came over for a champagne breakfast with the Wellington Apple group):

```
1  REM ENIGMA NOT DIVISABLE BY 0 OR 5
2  REM ENIGMA DIVISABLE BY 7,8,9,11
3  REM ENIGMA IS A MULTIPLE OF 5544
4  REM ENIGMA > 127512  (5544*23)
5  REM N HOLDS ENIGMA
6  REM C IS DIGIT SUM, WITH CARRY
10 DIM N(5): N(5)=1: N(4)=2: N(3)=7: N(2)=5: N(1)=1: N(D)=2: C=0

100 FOR I=0 TO 5:C=N(I)+(I/2+4) MOD 6+C/10: N(I)=C MOD 10: NEXT I: FOR
J=0 TO 4: FOR I=J+1 TO 5:IF N(J) MOD 5=0 OR N(I)=N(J) THEN 100:NEXT
I,J
110 FOR I=5 TO 0 STEP-1: PRINT N(I);:NEXTI:END
742896
```

THE PENGUIN PERSONAL COMPUTER COLLECTION


CREATIVE ASSEMBLER
How To Write Arcade Games
for the BBC Microcomputer Model B and Acorn Electron
Jonathan Griffiths
PENGUIN ACORN COMPUTER LIBRARY


How To Write
ADVENTURE GAMES
for the BBC Microcomputer Model B and Acorn Electron
Peter Killworth
PENGUIN ACORN COMPUTER LIBRARY

two books to
help you make
the most of
your BBC Micro

CREATIVE ASSEMBLER: How to Write Arcade Games for the BBC B and Acorn Electron.

A challenging, illustrated guide which introduces assembler instructions for the fast-moving and fun-packed arcade games.
$16.95

HOW TO WRITE ADVENTURE GAMES for the BBC B and Acorn Electron.

Particularly easy to follow instructions teach readers who have started programming in BBC BASIC how to create more complex adventure games.
$16.95

available from bookshops
and computer stockists

MENTAL ARITHMETIC GAME

G.C.F.

How many simple sums can you do in 60 seconds, using only whole and positive numbers (integers) in the range 1 to 100, using addition, subtraction, multiplication and division? Here is a game to let you find out.

This is the bare bones of a program for drill; 'frills' such as sound and graphics have been left out although even moving to a higher MODE will still allow plenty of memory for these.

DESIGN NOTES:

a) Should run as is on either Beeb or Electron.
b) No negative or fractional answers, also avoid multiply or divide by zero or an answer of zero.
c) All numbers to be one or two digits long (although '100' is also allowed).
d) each of the four arithmetic operators (+,-,*,/) should be tested.
e) use the conventional symbol for division, rather than '/', for clarity.
f) to avoid discouragement, correct answers bring a greater 'reward' of 10 points than the penalty for a wrong answer (1 point).
g) a small element of competition is introduced by keeping a high score.
h) use of special code and program techniques should be kept to a minimum.
i) to simplify typing in, single-letter variable names are used except where there is not an obvious one to choose (eg T% for Temporary variable, but op$ and time% are left in full)

PROGRAMMING NOTES:

To keep all the numbers between 1 and 100, including the answer (!) is not as easy as might seem before you start to program. You cannot simply select any two numbers in the range 1 to 100 and hope they will produce the type of answer you want. eg:

43 + 67 = 110 (too big)
43 - 67 = -24 (don't want a negative answer)
43 * 67 = 2881 (much too big!)
43 ÷ 67 = ?? (leads to fractional answers, or zero plus remainder)

For ease of discussion, the corresponding variable names will be used: A%, B, C% - in the first example above, A%=43 B%=67 C%=110. First we find out how to generate appropriate values of A% and B% for addition and multiplication.

For addition it is obvious A%=RND(99) and so B%=RND(100-A%) - but what if A%=99? We would then be asking for B% to be RND(1) effectively, which will not give an answer we want. See Coding note (e).

We can get the values for subtraction by noting that if C%=A%+B% then A%=C%-B%. This is why in the program PROCsub calls PROCadd and then swaps A% and C%!

For multiplication we want both of A% and B% to be greater than 1, and C% not to exceed 100. Therefore A%=RND(50), but calculating B% is trickier. Some possibilities:

If A% is greater than 33 then B% can only be 2, since we don't want to multiply by 1, and to multiply by 3 would make C% too big.

If A%=2 then B% can be in the range 2 to 50, or if A%=3 then B% has to be in the range 2 to 33.

Now 100 DIV2 is 50, and 100 DIV3 is 33, so that suggests perhaps B%=RND(100DIVA%). But that allows B%=1 again! How about B%=1+RND((100DIVA%)-1) Better, but we still have to avoid the possibility of RND(1) which would occur if A% was greater than 3. To avoid A% normally being larger than B%, we swap them 50% of the time (IF RND(2)=2...)

For division we used the multiplication PROC and swapped A% and C% just as for subtraction. Provided B% is not zero, which has been ensured in the program, C%=A%*B% then A%=C%/B%.

CODING NOTES:

a) Negative INKEY values - when you want the program to wait until a specific key is pressed, and upper or lower case isn't relevant - 'Press space bar to continue' you can:

REPEAT UNTIL INKEY(-99). The INKEY(-99) keeps testing the keyboard and will only let the program continue when space bar has been pressed, and no other key.

The same can be used for testing Yes or No - INKEY(-69) or INKEY(-86), with the added advantage of not having to worry if CAPS LOCK is on or off!

b) TIME - this measures time elapsed in hundredths of a second - which accounts for the number 60000 when we wanted 60 seconds from 'NOW'.

c) *FX15,1 - if anyone has typed something (or leaned on the keyboard accidently) this will clear if from the keyboard buffer.

d) LOCAL - this is used when the variable is only going to be used within the PROCedure; it is especially useful if there is a variable of the same name in the segment of code that calls the procedure and you don't want to corrupt that value of it.

e) RND - produces integers from 1 to whatever number specified (within reason and within parenthesis) - eg.to get any number up to 99 use RND(99) - but care should be taken if the argument (the value in the parenthesis) is a variable, as RND(0) and RND(1) have special meanings. The IF statements in lines 540 and 720 are there to avoid this problem.

f) VDU23 - used to define a character which looks more like the familiar division sign non-computer people expect.
g) CHR$ used to get the division character printed.

h) DIV this is the fastest way of dividing two integers, when you also want an integer for the answer (eg 7DIV2 gives the answer 3). It is also better documentation, as it tells the reader that you aren't interested in a 'real number' - a number that can have a fractional part - such as 2.5

VARIABLES:
A% and B% - first and second numbers in the sum
C% - Correct answer as calculated by the program (eg C%=A%+B%)
H% - High score
S% - current Score
R% - Reply the user gave in answer to the sum
T% - Temporary variable - only used locally; its value has no lasting significance.
time% - Time at which test is to end (approximately 60 seconds after start).
op$ - the OPeration symbol to be used (+, -, *, /) in a sum.

* * *

TABLE OF OVERLAPS

In any graphics mode, there are 32 lines of text on a screen. These are divided evenly down the Y-axis graphics range of 0 to 1023. (ie each character or line is 1024/32= 32 graphics units [points] high).

The character width varies according to the number of characters across the screen. For instance, Mode 0 has 80 characters across. 1280/80=16 so each letter is 16 graphic units wide.

Mode0	80 chars	16 graphic units wide
Mode1&4	40 chars	32 graphic units wide
Mode2&5	20 chars	64 graphic units wide

So a handy formula for converting TABX,Y to graphics co-ordinates is:
MOVE char_width*X, 1023-(Y*32)
char_width is the width in graphic units according to the Mode you are in.

```
   10 REM  MATH60
   20 REM  Game of Mental Arithmetic
   30 REM  By GCF 09-NOV-84
   40 MODE 6
   50 PRINT"You have 60 seconds to ans
wer"
   60 PRINT"as many arithmetic problem
s as you can"
   70 PRINT
   80 PRINT"RIGHT answers gain 10 poin
ts"
   90 PRINT"WRONG answers lose only 1
point"
  100 PRINT TAB(5,23)"Press SPACE BAR
to continue..."
  110 REPEAT
  120 UNTIL INKEY(-99)
  130 H% = 0
  140 REPEAT
  150 S% = 0
  160 time% = TIME + 6000
  170 REPEAT
  180 CLS
  190 PRINT"Score is ";S%;"  hi_score
is ";H%
  200 T% = RND(4)
  210 IF T% = 1 THEN PROC_add
  220 IF T% = 2 THEN PROC_sub
  230 IF T% = 3 THEN PROC_mul
  240 IF T% = 4 THEN PROC_div
  250 PRINT TAB(4,12);A%;" ";op$;" ";B
%;" =";
  260 *FX15,1
  270 INPUT" "R%
  280 IF R% = C% THEN S% = S% + 10 ELS
E S% = S% - 1
  290 UNTIL TIME > time%
  300 CLS
  310 IF S% = H% THEN PRINT"You have e
qualled the previous hi score of ";H%
  320 IF S% < H% THEN PRINT"You need b
etter than ";S%;" to beat ";H%
  330 IF S% > H% THEN H% = S%: PRINT"Y
ou have increased the hi score to ";H%
  340 PRINT TAB(4,23)"Another game Y/N
 ?"
  350 REPEAT
  360 UNTIL INKEY(-69) OR INKEY(-86)
  370 UNTIL INKEY(-86)
  380 CLS
  390 *FX15,1
  400 END
  410
  420 DEF PROC_sub
  430 LOCAL T%
  440 PROC_add
  450 op$ = "-"
  460 T% = A%
  470 A% = C%
  480 C% = T%
  490 ENDPROC
  500
  510 DEF PROC_add
  520 op$ = "+"
  530 A% = RND(99)
  540 IF A% = 99 THEN B% = 1 ELSE B% =
RND(100 - A%)
  550 C% = A% + B%
  560 ENDPROC
  570
  580 DEF PROC_div
  590 LOCAL T%
  600 PROC_mul
  610 VDU23,224,&18,&18,&00,&FF,&00,&1
8,&18,&00
  620 op$ = CHR$(224)
  630 T% = A%
  640 A% = C%
  650 C% = T%
  660 ENDPROC
  670
  680 DEF PROC_mul
  690 LOCAL T%
  700 op$ = "*"
  710 A% = 1 + RND(49)
  720 IF A% > 33 THEN B% = 2 ELSE B% =
1 + RND((100 DIV A%) - 1)
  730 IF RND(2) = 2 THEN T% = A%: A% =
B%: B% = T%
  740 C% = A% * B%
  750 ENDPROC
```

OSWORD FROM BASIC!

GCF

INTRODUCTION

You can use many of the Operating System routines without having to know anything about machine code (I do - I've had enough experience of writing machine code for main-frame computers at work that I will not use it on the Beeb until I am absolutely forced to! - despite the "ease" of the Beeb's assembler)

In this article I want to show what is involved in setting up and using OSWORD functions. For a list of some of them, refer p.458 User Guide, and chapt. 8 Disc U.G. (don't be put off by its title "Using the filing system in assembler" - it can all be done from Basic).

To make it easier to grasp, the following example uses an OSWORD function which does exactly the same thing as a 'well-known' Basic command POINT.

EXAMPLE:

All OSWORDS follow the same general pattern:

a) grab some space - the amount depends on the particular OSWORD. This is called the PARAMETER BLOCK. It is used to pass data to and from the OSWORD routine (eg DIM block% 4 - the space between variable name and number is needed; here it reserves 5 bytes: 4+1) There is no harm in reserving more space than you need for the Parameter Block.

b) point to it - the OSWORD routine needs to know where the parameter block starts. This is done by splitting the starting address into 2 bytes and placing them in X% and Y%.
eg X%=block%MOD256:Y%=block%DIV256

c. select function - put the function number in A% so the OSWORD handler knows which routine to activate. eg A%=&09. It is tradition to use hex; this reads the logical colour of a pixel.

d. Insert parameters - it is now necessary to poke the parameters into this block (chapt. 39 U.G. for indirection operators). eg in this example we must pass the graphics co-ordinates of the pixel we are interested in.

e. call it - actually you call a bit of code which jumps somewhere else depending on a vector. eg CALL &FFF1
Understanding these technicalities is not essential to use OSWORD successfully.

f. extract results - this stage is not required for some OSWORD functions such as "write internal timer" (A%=&04), and varies wildly for others eg. extract the logical colour of the pixel.

g. to reuse the same OSWORD function only steps d, e and f need repeating.

```
 10 REM  OSWORD9
 20 REM  A direct equivalent of the BASIC command "POINT"
 30 REM  See USER GUIDE pages 322 & 462
 40 MODE5
 50
 60 REM  a) Grab some space
 70 DIM block% 4
 80
 90 REM  b) Point to it
100 X% = block% MOD 256
110 Y% = block% DIV 256
120
130 REM  c) Select function
140 A% = &09
150
160 REM  (select pixel to test)
170 H% = RND(1279) :REM horizontal coordinate
180 V% = RND(900)  :REM vertical coordinate
190 GCOL0,2        :REM make it yellow
200 PLOT 69,H%,V%  :REM plot the point
210
220 REM  d) Insert parameters
230 ?block% = H% MOD 256
240 block%?1 = H% DIV 256
250 block%?2 = V% MOD 256
260 block%?3 = V% DIV 256
270
280 REM  e) Call it
290 CALL &FFF1
300
310 REM  f) Extract result
320 logical_colour% = block%?4
330
340 REM  (print result - it should be 2)
350 PRINT logical_colour%
360 END
```

CONCLUSION:
Yes, using POINT from Basic is much more convenient, but there are other, more useful OSWORD functions which have no equivalents in Basic. One such extracts the bit pattern of characters (A%=&0A) in a way that is Tube-compatible. People with disc systems can also use an OSWORD (A%=&7F) to read ANY sector on ANY drive.

I have prepared articles on the following OSWORD functions:
A%=&0A - Display Character Matrix
A%=&0B - Find Actual Colour
A%=&7F - Disc Sector Verify

EDIT CHARACTERS— II

Neville T. Ch'ch

Last time I described a short procedure which allowed you to design a used-defineable character on the screen, and which could then be automatically poke it into memory, hence defining it for you.

Now let's look at how we can pull a character out of memory, edit it on the screen and then restore the edited version back in memory. This is a little more complicated. However once we have the information and displayed the character, we can use the procedure developed last time to do the editing and restoring.

We know that characters are stored in sections of 8 bytes in Page &C00 to &CFF; the first 8 bytes corresponding to CHR$224, the next 8 to CHR$225 and so on up to CHR$255.

So a formula to find the first byte of any character is:

(char-224)*8+&C00

Now here is a routine to pull the information out of memory and display it in block form on the screen:

```
10 MODE7:INPUTTAB(10,10)"Edit which character ",char
20 MODE7
30 FOR I=0 TO 7:value=?((char-224)*8+&C00+I)
40 FOR J=0 TO 7
50 IF value>=2^(7-J) THEN ?(HIMEM+J+I*40)=255:value=value-2^(7-J)
60 NEXTJ:NEXTI
```

That is all there is to it! We input the character to be editted, store it in the variable 'char'. We then use two loops. The outside loop (I in line 30) steps through each of the 8 rows of a character. As each row is represented by one byte in memory, this loop reads the appropriate byte each time round using:
?((char-224)*8+&C00+I) -see Oct Beeblet.

Line 50 checks to see whether this block should be printed on screen by comparing it to the column value. Hence at the beginning of the loop when J=0, 2^(7-J) =2^7 or 128, whereas at the end J=7 so 2^(7-J) = 2^0 = 1

If the block should be printed, the ASCII value of a solid block in Mode 7, CHR$255, is poked into the correct location of screen memory. This displays the character.

The whole procedure, using the previous article's routines and the one above is as follows:

```
   10 MODE7:INPUTTAB(10,10)"Edit which character ",char
   20 MODE7
   30 FOR I=0 TO 7:value=?((char-224)*8+&C00+I)
   40 FOR J=0 TO 7
   50 IF value)=2^(7-J) THEN ?(HIMEM+J+I*40)=255:value=value-2^(7-J)
   60 NEXT,
   90
  100 X=0:Y=0:REPEAT:PRINTTAB(X,Y);:Q=GET:X=(X+(Q=90)-(Q=88))MOD8:Y=(Y+(Q=58)-(Q
=47))MOD8:X=X-(X(0)*8:Y=Y-(Y(0)*8:?(HIMEM+X+Y*40)=Q=32:UNTILQ=83
  110 I=0:FOR Y=0TO7:C=0:FOR X=0TO7:C=C+(2^(7-X) AND ?(HIMEM+X+Y*40)=255):NEXT:P
RINTTAB(27,Y);C:?((char-224)*8+&C00+I)=C:I=I+1:NEXT
```

Line 10 asks for a character, line 20 displays it.
line 30 provides on-screen editing in this way:
Z (left)   X (right)   : (up)   / (down)   S (finished)   <space> to print a block.
line 40 restores the (new) character to memory.

Well, what are you waiting for?... Go to it!!!

COPROCESSOR SCREEN DUMP

thanks to Neil G for explaining

If you have the source code for a machine-code screen dump which works for a Beeb without an active co-processor then it can also be used with an active one.

Change the Load Address so that it will load at &1F00 in the I/O processor and set the execution address accordingly. If this has been done correctly then doing *INFO on your disc system should show that the load and execution addresses are prefixed by "FF". This means they will load and run in the I/O processor. The reason for the particular address chosen is to avoid corrupting parts of the I/O processor used in handling the Tube as well as the expanded User-defined character set area - not to mention the disc buffer areas...

Say your dump routine is called CODUMP. You can activate it from a Basic program in the co-processor by *RUN CODUMP. This will fetch the program from disc, produce the screen dump, then continue with the Basic program.

WANTED

An explanation of how to run CODUMP in the I/O processor from a Basic program running in the co-processor, once it was loaded.

A HEX ON THAT CHARACTER!

Anne F.

I can't add up! (or, not as accurately as I used to) So to expect me to add up 128+32+8+1, or some such combination to arrive at a User Defined Character and get it right, is really asking a bit much. The calculator always needs recharging, the computer is laborious, and anyway, am I sure that I've got the right column to add in in the first place?

I work in hex (gulp) but cheat (hurrah!) - with a look-up table beside me. Each combination of squares has a single value. For instance:

□□□■ this is "1"
□□■□ this is "2"
□■■■ this is "7" or 4+2+1
■■□□ this is "C" or 8+4 = 12
□■■■ this is "7" or 4+2+1

My character is staring me in the face, all neatly divided into squares, so it makes sense to take advantage of this and forget the arithmetic. Great, marvellous, but the examples above only deal with 4 squares, and a character has 8. Isn't there going to be some arithmetic anyway? No - you just put the two numbers side by side and the computer is happy. For instance:

■■□□|□□■■ = C3 (properly &C3 or 1+2+64+128?)

Here is a look-up table

□□□□	0
□□□■	1
□□■□	2
□□■■	3
□■□□	4
□■□■	5
□■■□	6
□■■■	7
■□□□	8
■□□■	9
■□■□	A
■□■■	B
■■□□	C
■■□■	D
■■■□	E
■■■■	F

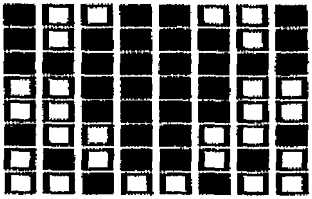
and an example

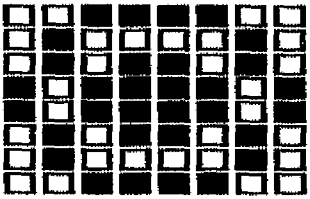
□■□■	□■■■	57
□■□■	□□■□	52
□■□■	□□■□	52
□■■■	□□■□	72
□■□■	□□■□	52
□■□■	□□■□	52
□■□■	□■■■	57
□□□□	□□□□	00

By comparing each row (half at a time) with the pattern in the table, you can quickly arrive at the number that represents that row.

Using hex consistently soon gives you a feel for a shape too. Did you notice with "C3" that 3 is the reverse of C?
Or that in the full character above, the top line is repeated at line 7, that moving in towards the centre from lines 1 and 7 the rows are the same until line 4 splits them?

Try converting the following shapes to hex:





Taking the first one,

■□□■■□□■ means &99
■□■■■■□■ means &B6
■■■■■■■■ means &FF

... and so on.

By avoiding arithmetic, and having a table handy to compare the shapes with their value, it is easier to find an error in a definition and correct it. Also when typing in definitions it is also possible to omit a number, or add in an extra one. Using hex, such a definition stands out like a sore thumb, because all the definitions look as long on the computer screen. Spot the odd one below for instance:

```
VDU23,240,&99,&B6,&FF,&3C,&3C,&99,&5A,&24
VDU23,241,&3C,&42,&5A,&BD,&BD,&5A,&42,&3C
VDU23,242,&00,&01,&0A,&1C,&3E,&7C,&18,&10
VDU23,243,&24,&25',&24,&AE,&35,&AE,&77
VDU23,244,&00,&00,&00,&10,&38,&58,&10,&28
```

(Add &AA to the end of CHR$243 to complete the pattern.)

AMPERSAND:
This is the word to describe the character that looks like & and means a number following it is a hexadecimal number. It is on the keyboard on top of the number 6 key - to get it, press SHIFT and 6. A

quicker way to set up lots of character definitions is to define a red key to produce the comma ampersand that separates each number:

`*KEY4¦,&`

For some reaon you need the broken-line character at the start of the definition, otherwise the comma doesn't appear.

◆

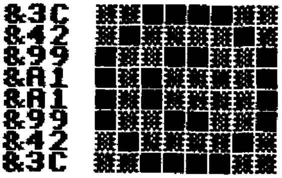
VIEW KEYS

Don A ToKoroa

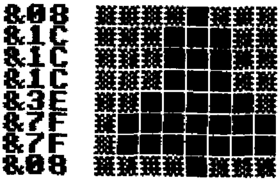
To make use of the red user-defined keys in Command Mode, set them in the normal way, type *FX228,1 and access their contents by holding CTRL+SHIFT while pressing the U-D key. This overcomes the problem of having the text mode reset the user-defined keys to its own purposes when the command/text mode toggle (ie ESCape key) is used.

◆

Neil wanted a Hi-Res Logitec screen dump. No joy from our members! Jeff Aughton's excellent book "Invaluable Utilities for the BBC Micro" pub. Pan was much more use. Neil also recommends the "Graphics Aid" program, to which he has added dotted line, arc and ellipse drawing functions as well as the screen save and printer dump utilities found in the book.



© is CHR$(225)



♣ is CHR$(226)

DISPLAY CHARACTER MATRIX

GCF

This program finds the bit patterns used to define the various characters and displays the 8x8 pixel matrix in an expanded form. On the left hand side the hex values required to define the particular character using VDU23 are given. On the other side, the character is printed in normal size, with its ASCII value in decimal. All the characters from ! (pling) to ~ (tilde) are displayed in turn, with a 1-second delay between each.

The program could be used to help understand how existing characters are defined, before attempting to define your own. For most characters the left line and bottom row are left blank. The only alphabetic exceptions being gjpqy. By using OSWORD with A%=&0A, this program will work across the Tube. By changing the start and finish value you can display the current values (definitions) of the User-defined characters for the ASCII range 224 to 255. (eg 100 FOR asc%=224 TO 255)

```
 10 REM  OSWORDA
 20 REM  This 'reads' a character definition.
 30 MODE6
 40 VDU23,224,&55,&2A,&55,&2A,&55,&2A,&55,&00
 50 VDU23,225,&7F,&7F,&7F,&7F,&7F,&7F,&7F,&00
 60 DIM block% 8
 70 A% = &0A
 80 X% = block% MOD 256
 90 Y% = block% DIV 256
100 FOR asc% = 33 TO 126
110   ?block% = asc%
120   CALL &FFF1
130   PRINT '''
140   FOR I% = 1 TO 8
150     byte% = block%?I%
160     PRINT ' "&" + RIGHT$(STR$~(&100 + byte%),2) + " ";
170     FOR J% = 7 TO 0 STEP -1
180       IF byte% AND 2^J% THEN VDU225 ELSE VDU224
190       NEXT J%
200     IF I% = 4 THEN PRINT SPC(4) CHR$(asc%) " is CHR$(";asc%")";
210     NEXT I%
220   wait = INKEY(100)
230   NEXT asc%
240 END
```

CROSSWORD

Neville T.   Ch'ch

All the words used are terms related to the Beeb or Elk; in fact all the answrs can be found by scanning the Users Guide.

ACROSS:

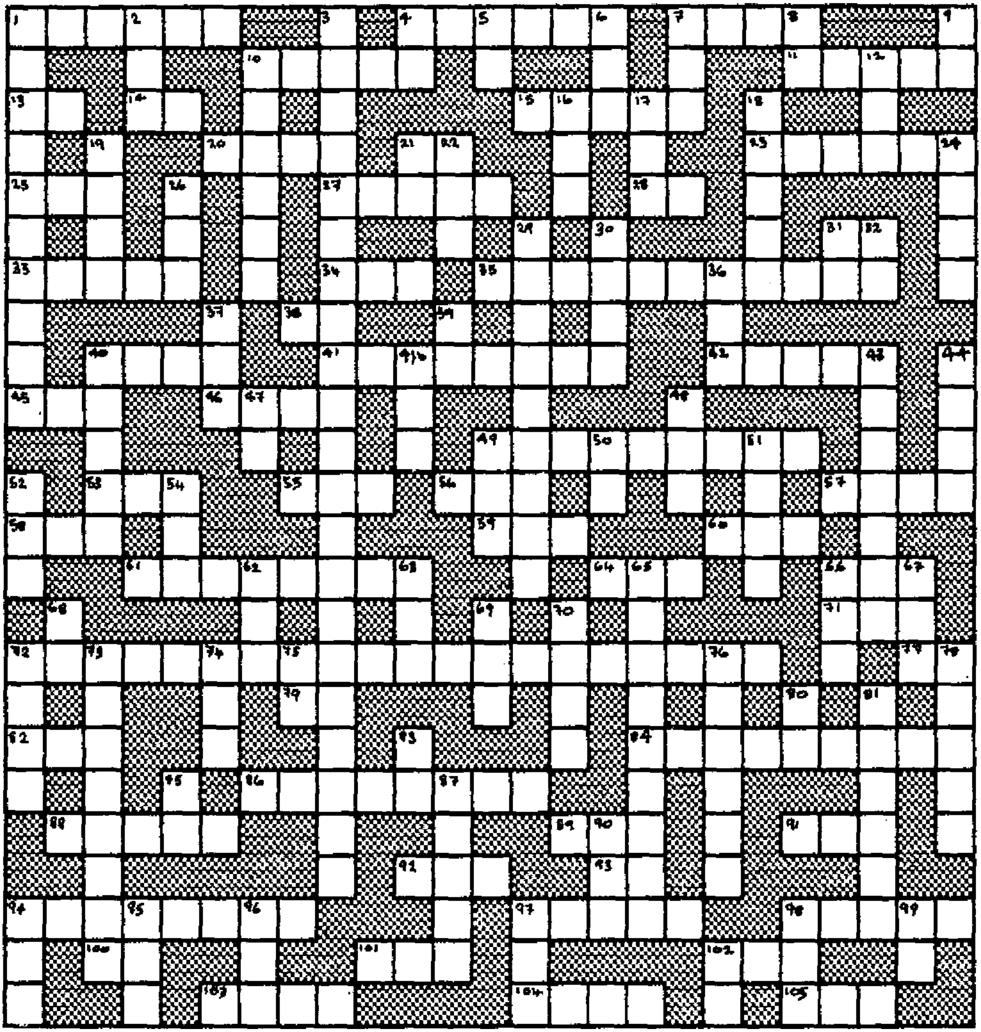
1 Government takings for wrong-doing? - also error 16 (6)
4 Error with huge numbers (3,3)
7 Reaction to a red light (4)
10 Event Vector (5)
11 Substring search (5)
13 Minimum abbreviation of RIGHT$ (2)
14 Hex ASCII of '?' in MODE 7 (2) clue: It's >120
15 Parameters of 3-8 of 73 down (5)
20 Not considered viable by structured programming buffs (4)
21 Hex 174 (2)
23 Find a file to read from (6)
25 Assembly mnemonic that puts a value in the Accumulator
27 What do a cassette recorder and car have in common? a *call too (5)
28 abbreviation for CHR$30 (2)
31 BASIC conditional execute (2)
33 Modulator demodulator (5)
34 Request for servicing by the 6502 (3)
35 ?,! and $ are all examples of these operators (11)
38 Abbreviation for the VDU code to move the cursor up a line (2)
40 Bank on the cassette! (4)
41 The 11 screen memory mode (8)
42 Access speaker output (5)
45 Powerful raising (3)
46 OSWRCH address (4)
49 Quarrels (9) also an error
53 MC error/listing suppress or OS call (3)
55 abbreviation of 3 down (3)
56 In some cases, this is what a monitor is called... in others... (3)
57 Visit a friend or execute some machine code (4)
58 Mistaken line! (3)
59 The heart of a computer - ours is a 6502 (3)
60 VDU 0 (abbr) (3)
61 Digital opposite (eg in joysticks) (8)
64 &FD00-&FDFF (3)
66 6522 chip (3)
71 timber-chopping tool mis-spelt (3)
72 *FX19 waits for it (5,15) P.S. use NZ spelling 's' not 'z'
77 abbr for the RETURN-generated character (2)
79 Hex for ASC of a teletext star (2) see 14 across
82 Whole number part of division (3)

84 The 6502 has 3: A,X, Y (9)
86 opposite to 81 down (8)
88 reset all variables and arrays (5)
89 Put A Register in the Stack (Assembly mnemonic) (3)
91 Disc interruption (3)
92 Assembly comparison mnemonic (3)
93 Hex value of a semi-colon's ASC times the number of Analogue/Digital Converters on Model B Beeb (2)
94 activates a Keyboard LED (4,4)
97 The smallest graphics unit (5)
98 Renumber error, dope! (5)
100 Using VDUs hex value of (code to switch on page mode * code for defining text colour) (2)
101 Not so bright (3)
102 Size of a disc file (3)
103 CHR$69+66 down (4)
104 Grab a variable from a line in the program (4)
105 Spin left (Assembly mnemonic) (3)

DOWN

1 opposite of page mode (CHR$15) (6,4)
2 moves text cursor (3)
3 Motorola's video chip in full (11,5,5)
4 OS command to move the monitor display (2)
5 conditional branch for GOTO, GOSUB (2)
6 wait for input (3)
7 VDU abbr for CTRL-A (4)
8 Mathematical food? (2)
9 rowboat conditional (2)
10 Community sharing of BBCs and peripherals (6)
12 circular crime (3)
16 chops those decimals! (3)
17 opposite of 67 down (3)
18 ALMOST duplicates POS (5)
19 Data Transfer Rate (4)
21 name for CHR$64 (2)
22 a one-of-a-Kind 9 down (3)
24 Repeat FOR (4)
26 Model B has slots for 5 of these (3)
29 the 6502 halts its present task for one of these (9)
30 look at a program on the screen (4)
31 input/output abbr (2)
32 subroutine Keyword that only returns one value (2)
36 wipe the screen clean (3)
37 always comes before you write a PROC or FN (3)
39 memory location in page zero holding the current value of 18 down (hex) (2)
40 OS command used in program merging (5)

41b common timber? (3)
43 opposite of STATIC (7)
44 choose a colour to draw with (4)
47 VDU abbr for defining text window (2)
48 an entirely optional keyword; need never be used! (3)
49 joystick port or assembly addition mnemonic (3)
50 abbr for your guide (2)
51 logical opposite of zero (4)
52 radial opposite (3)
54 circular shade of brown! (3)
62 store a value in Y (assembly mnemonic) (3)
63 the number associated with 58 across (3)
65 connection between 2 pieces of equipment (9)
66 decimal value of a string (3)
67 gives you the ASCII code (3)
68 minimum abbr for the keyword reading the internal clock (2)
69 finish your program here (3)
70 YOU (also 50 down abbr's first word) (4)
72 OSARGS hex address (4)
73 letter storage - Sound advice! (8)
74 used to print multiple spaces
75 Not Applicable (2)
76 OS call at &FFD1 (6)
78 BREAK is a soft one; CTRL-BREAK a hard one! (5)
80 the 2 letters common to all calls in memory in the &FF00-&FFFF range (2)
81 type of data transfer, but not what you would eat for breakfast (6)
83 minimum abbr for a versatile command that even fills triangles! (2)
85 hex value of *FX call to insert a character in the keyboard buffer (2)
87 start of variable storage space (5)
90 base 16 numbers (3)
94 graphics equivalent of 36 down (3)
95 assembly mnemonic to set interrupt (3)
96 assembly mnemonic to wipe the evidence of an overflow (3)
97 where we're up to in a disc file (3)
98 opposite of 66 down (3)
99 non-synthetic timber? (2)
102 hex of *FX call to disable the ESCAPE key (2)



GRAPHS or...
I'M SICK OF NUMBERS

Anne F.

There comes a time when lists of numbers become meaningless, just so many digits dripping off your hands (and toes in summer). What is needed is a pictorial or graphical way of showing quickly what such lists of numbers look like, and what the relationship between them all is, since that can be more meaningful than the specific numbers. (Which gives you a better feel for the time?: a digital clock saying "9.57" or an analogue one saying "almost 10 o'clock! Time for bed!")

The program here is an extension of the one calculating a woman's age. As I was using it, the data was all read in from a disc file. However here it is presented as Data statements. It draws two graphs. Again, it looks like a long program, but that is because it is not a "bare bones" program ie one that says in the explanatory text "add sound, graphics, colour..." though I must confess I've ignored sound completely here. I have, however, gone to quite a bit of trouble to polish it, and done things which wouldn't be apparent unless they had not been done - if then. If you have your own set of dates to use, use them to replace those in lines 1650 on. Also change the value of count% in line 15.

<u>Graph 1: PROCmonthgraph</u>
A bar graph: at a glance you can see what is the most popular month for people to be born in. You may recall the graph on p.40 U.G., done in Mode 2, with lots of colours available. I don't like this mode because the characters are too fat. Instead I've taken advantage of the 'Impossible Colours' routine in Feb'84 Beeblet, and used Mode 1.

<u>MOD colours</u>
MOD is a very useful command for when you can't be bothered specifying each time which in a series of numbers, colours etc, you want if you are wanting to cycle through that series:

```
570 IF numberinmonth%(I%) > 0 THEN
      PROCbox(I%MOD4,I%DIV3+1,X%+12,Yoffset%,X%+width%,Y%)
```

IF anyone was born in the month I%, draw a box. Use for the first colour, the remainder of I% divided by 4. This means the first colour will cycle round 4 possibilities in order:
I% = 4 so I%MOD4 = 0; I% = 11 so I%MOD4 = 3 (ie 3 is the remainder when you divide 11 by 4).
The second colour is I%DIV3+1. 1 is added to the remainder, so this colour will never be colour 0.

OFFSETS...

When I originally wrote this program, I was defining graphics windows to produce the boxes. This implied havoc if I used the VDU29 command to set the origin of the graph axes. So I faked this by Yoffset% which starts the graph so far up from the bottom of the screen, and Xoffset% which starts the horizontal axis in from the left of the screen.

...AND OTHER VARIABLES

Designing a screen is only approximate, so it made sense to make every possible co-ordinate a variable - especially when they interact with other co-ordinates - as it is easier to change any one, and know that all occurrences of it will also be changed. For instance, width% in PROCmonthgraph. This variable is how wide each box is to be. It is used not only in making sure each month's box doesn't overwrite a previous one etc, but also to centre the month name under the relevant box:

```
590 MOVE X%+(width%-32)/2,text%:PRINTmonthname$
```

ie however wide the box is, subtract from it the width of the character representing the month name, and divvy up on each side of the character the remaining space.

When deciding what to put where on the screen, you are working in generalities first, so you might as well make the program as general as possible - look how I was able to lift PROCbox out of another mag. I only omitted one line, and changed the name from PROC_M10. This leads us on to...

PROCyeargraph

This graph takes advantage of PROCproduce_sorted_list which sets up an array of pointers which tell the computer what order to pull people out of the other arrays to be in order of age . I gave up on sorts when my Apple Pascal tutor set us an exercise to sort 10 characters into order without using arrays. So ask Gavin how this one works (with arrays).

I don't know how many people are going to be in your DATA statements, but I intended them to be evenly spread up the screen - 4 people would be displayed over the same area as 20. In practice the program can only cope with 32 people before their names start to overwrite each other:

count% is the number of people PROCname has to print up. Ywidth% is the amount of space (Y co-ordinates) available after reserving space for the title and the bottom offset (1024 -Yoffset% -32 which is the height of characters for the top line)

By dividing Ywidth% by count% each person will be spaced evenly.

This same principle is applied to setting out the age range across the screen: find out the oldest age and use Xwidth% to space the age range evenly across the screen.

PROCage_points

This provides some reference points - it prints age across the screen in tens - 10,20,30... up to max_age%. On most graphs, such labels are along the bottom (as month names in the first graph). However I felt they would be a little more meaningful halfway up the screen instead, since a splodge is placed at the age of each person, rather than a line drawn connecting each point.

COLOUR

Besides that of the first graph, which is a special effort. Notice the colours chosen in the second: yellow names and splodges, purple reference points - on my monitor at least, this gives the illusion on the references being further back, as a background in front of which individual people are placed.

OTHER PEOPLES' GRAPHS...

I was typing up a load of material from other members for the Beeblet Editor, and came across two little programs, both doing pretty much the same thing, but graphing the results in different ways. On one, the Editor had written "Steven, write some blurb". On the other, Anton had written the following:

This program is simply a way of illustrating the areas of BASIC that contain statements. A permanent sink of 13 is put on V% and an increase of 100 every time an ASCII value is found either in the BASIC ROM (&8000 to &BFFF) or in the OS (&C000 to &FFFF).

What both graphs are doing then, is chugging through these ROMs, looking for ASCII characters, and displaying on screen how many of each there are. This can be looked on in one of two ways:
1) An intellectual exercise
2) using the computer's ROMs as a convenient source of data (anyone want to type in 32K of data for analysis?)
In Steven's program, one item goes off the screen (scaling?) - it actually represents the number of CHR$32 (spaces) found.
If I've misrepresented either of them, apologies, but
1) you should have written something or more,
2) been available when I prepared this.

FINALLY...

The limitation of all four graphs is that they only show points based on two items (number in month, per month; name, age; number of times, ASCII value of each). When more than this is required to be visualised or compared, there are other methods:
Pie charts (as in Jan '83 and March '83 Beeblets - copies still available $2 each from the Secretary)
and something lurking on one of our discs under the filename "GERPH". But that's for another day........

```
   10 REM *** CUMULATIVE FREQUENCY TABLE ***
   20 REM ***    OF ASCII IN BASIC ROM   ***
   30 REM ***     by Steven, March '84   ***
   40
   50 MODE1
   60 DIM T%(255)
   70 VDU23,1,0;0;0;0;:REM  cursor off
   80 REM the above line is for OS 1.0 - OS 0.1 should
refer U.G.p.77
   90 VDU29,150;0;

100 PRINTTAB(0,1)"1000";TAB(2,31);"0";TAB(37,31)"255";TAB
(0,0);CHR$11
  110 MOVE 1020,0:DRAW -7,0:DRAW -7,1023
  120 FOR D%=&8000 TO &BFFF STEP &100:REM look at BASIC
ROM
  130   FOR P%=0 TO 255
  140     M%=D%?P%
  150     T%(M%)=T%(M%)+1
  160     NEXT
  170   PRINTTAB(16,0)"PAGE &";~D%
  180   FOR S%=0 TO 1020 STEP 4
  190     MOVE S%,0:DRAW S%,T%(S%DIV4)
  200     NEXT S%
  210   NEXT D%
  220 VDU23,1,1;0;0;0;:REM cursor on
  230 PRINTTAB(16,2);
  240 END
```

```
   10 REM ***  ASCII SEARCH PLOT  ***
   20 REM *** by Anton March '84  ***
   30
   40 MODE1
   50 VDU29,0;200;
   60 B%=32
   70 H%=8192 DIV B%
   80 FOR I%=1 TO 4
   90   MOVE I%*H%,800:DRAW I%*H%,-200
  100   NEXT I%
  110 MOVE 4*H%,0:DRAW 0,0
  120 V%=0
  130 FOR M%=&8000 TO &FFFF
  140   N%=?M%
  150   IF N%>47 AND N%<91 THEN V%=V%+100
  160   V%=V%-13
  170   PLOT69, (M%-&8000) DIV B%, V% DIV 100
  180   NEXT
  190 END
```

```
   10 ON ERROR VDU4,26,20:REPORT:PRINT
" @ line ";ERL:STOP
   20 REM  produce some graphs based o
n "Womans age" (Beeblet Nov'84)
   30 REM  by Anne F
   40 REM  Sept '84
   50
   60 MODE1
   70 PROCinitialise
   80 PROCmonthgraph
   90 PROChold
  100 PROCyeargraph
  110 PROChold
  120 END
  130
  140 DEF PROCinitialise
  150 thisyear%=1984
  160 count% = 15
  170 DIM name$(count%),year%(count%),
month%(count%),day%(count%)
  180 DIM numberinmonth%(12) : REM use
d for PROCmonthgraph
  190 DIM position_in_list%(count%) :
REM used in PROCyeargraph
  200 DIM monthname$(12)
  210 PROCfillarrays
  220 VDU23,224,&08,&1C,&2A,&08,&08,&0
8,&08,&08:REM up arrow
  230 VDU23,225,&81,&DB,&E7,&DB,&99,&3
C,&7E,&00:REM  angel
  240 VDU23,226,&08,&08,&08,&08,&08,&2
A,&1C,&08:REM down arrow
  250 ENDPROC
  260
  270 DEF PROCfillarrays
  280 RESTORE 1640
  290 FOR I% = 1 TO count%
  300   READ name$(I%),year%(I%),month
%(I%),day%(I%)
  310   NEXT I%
  320 REM set up array for PROCmonthgr
aph
  330 FOR I%=1 TO count%
  340   numberinmonth%(month%(I%)) = n
umberinmonth%(month%(I%)) + 1
  350   NEXT I%
  360 RESTORE 370
  370 DATA J,F,M,A,M,J,J,A,S,O,N,D
  380 FOR I% = 1 TO 12
  390   READ monthname$(I%)
  400   NEXT I%
  410 PROCproduce_sorted_list
  420 ENDPROC
  430
  440 DEF PROCmonthgraph
  450 LOCAL I%,X%,Y%,Xoffset%, Yoffset
%, width%, text%
  460 Xoffset% = 64
  470 Yoffset% = 256
  480 width%=88
  490 text%=200
  500 VDU19,3,4,0,0,0:REM blue not whi
te
  510 PRINTSPC4;"NUMBER OF PEOPLE BORN
 EACH MONTH"
  520 REM calculate and plot bar graph
s
  530 VDU5
  540 FOR I% = 1 TO 12
  550   X% = I% * width% + Xoffset%
  560   Y% = numberinmonth%(I%) * widt
h% + Yoffset%
  570   IF numberinmonth%(I%) > 0 THEN
 PROCbox(I%MOD4,I%DIV3+1,X%+12,Yoffset
%,X%+width%,Y%)
  580   GCOL4,0
  590   MOVE X%+(width%-32)/2,text%:PR
INTmonthname$(I%)
  600   NEXT I%
  610 REM put numbers up the Y-axis
  620 FOR I%=0 TO 7
  630   Y%=I%*width%+Yoffset%
  640   MOVE 100,Y%-32
  650   GCOL0,3
  660   IF I%>0 THEN PRINT;I%
  670   GCOL0,0
  680   MOVE 100,Y%
  690   DRAW1279,Y%
  700   NEXT
  710 VDU4
  720 ENDPROC
  730
  740 DEF PROCyeargraph
  750 LOCAL I%,X%,Y%,Xoffset%, Yoffset
%, Xwidth%, Ywidth%, max_age%
  760 Xoffset% = 256
  770 Yoffset% = 64
  780 Xwidth% = (1170-Xoffset%) / (thi
syear%-year%(position_in_list%(count%)
))
  790 Ywidth% = 920/(count%-1)
  800 max_age%=thisyear%-year%(positio
n_in_list%(count%))
  810 VDU20:REM reset logical colours
to normal
  820 VDU19,1,5;0;:REM purple not red
or blue
  830 PRINTSPC9;"PEOPLE IN ORDER OF AG
E"
  840 VDU5
  850 PROCage_points
  860 PROCgraph_points
  870 PROCnames
  880 VDU4
  890 ENDPROC
  900
  910 DEF PROCgraph_points
  920 GCOL0,2
  930 MOVEXoffset%,Yoffset%
  940 FOR I%=1 TO count%
  950   X% = (year%(position_in_list%(
1))-year%(position_in_list%(I%))) * Xw
idth% + Xoffset%
  960   Y% = (I%-1) * Ywidth% + Yoffse
t%
  970   MOVE X%,Y%:VDU225
  980   NEXT I%
  990 GCOL0,3
 1000 ENDPROC
 1010
 1020 DEFPROCnames
 1030 GCOL0,2
 1040 FOR I%=1 TO count%
 1050   Z%=(I%-1)*Ywidth%
 1060   MOVE0,Z%+Yoffset%
 1070   PRINT;name$(position_in_list%(
I%))
 1080   NEXT I%
 1090 GCOL0,3
 1100 ENDPROC
 1110
 1120 DEF PROCage_points
 1130 GCOL0,1
 1140 FOR I%=10 TO max_age% STEP 10
 1150   MOVE Xwidth%*I%+Xoffset%,512+4
8:VDU224
```

```
 1160   MOVE Xwidth%*I%-16+Xoffset%,51
2:PRINT;I%
 1170   MOVE Xwidth%*I%+Xoffset%,512-4
8:VDU226
 1180   NEXT I%
 1190 GCOL0,3
 1200 ENDPROC
 1210
 1220
 1230 DEF PROCproduce_sorted_list
 1240 REM this PROC written by GCF
 1250 REM fill array with order of age
(year born)
 1260 LOCAL I%,J%,min%
 1270 LOCAL T%
 1280 FOR I% = 1 TO count%
 1290   position_in_list%(I%) = I%
 1300   NEXT I%
 1310 FOR I% = 2 TO count%
 1320   min% = I% - 1
 1330   FOR J% = I% TO count%
 1340     IF year%(position_in_list%(J
%)) > year%(position_in_list%(min%)) T
HEN min% = J%
 1350     NEXT J%
 1360   IF min% > I% - 1 THEN T% = pos
ition_in_list%(I% - 1): position_in_li
st%(I% - 1) = position_in_list%(min%):
 position_in_list%(min%) = T%
 1370   NEXT I%
 1380 ENDPROC
 1390
 1400 DEF PROChold
 1410 *fx15,1
 1420 PRINTTAB(8,31)"Press SPACE to co
ntinue";
 1430 REPEAT UNTIL INKEY(-99)
 1440 CLS
 1450 ENDPROC
 1460
 1470 DEF PROCbox(C1%,C2%,LX%,BY%,RX%,
TY%)
 1480 REM from Beeblet Feb '84 "Imposs
ible Colours" p.10
 1490 LOCAL CA%,CB%,T%,Y%
 1500 CA%=C1%
 1510 CB%=C2%
 1520 FOR Y%=BY% TO TY% STEP 4
 1530   GCOL 0,CA%
 1540   MOVE LX%,Y%
 1550   PLOT 21,RX%,Y%
 1560   GCOL 0,CB%
 1570   PLOT 21,LX%-4,Y%
 1580   T%=CA%
 1590   CA%=CB%
 1600   CB%=T%
 1610   NEXT Y%
 1620 ENDPROC
 1630
 1640 DATA Adrienne,1932,4,11,Margaret
,1958,7,23,Arfle,1980,8,2,Barfle,1980,
8,2,Gloop,1980,8,2,Doreen,1943,6,18,Jo
anne,1943,3,30,Chris,1966,10,24
 1650 DATA Stephanie,1959,10,3,Theresa
,1921,12,26,Wendy,1948,9,20
 1660 DATA Mary,1964,1,4,Julia,1953,2,
13,Lynn,1950,5,20,Greta,1920,11,30
```

SHALL I COMPARE THEE WITH THE LOWER CASE?

Joe (Hamilton) offers the following hint:
When running programs and testing input given, these *FX calls save checking for upper or lower case. They are useful when you aren't worried which is correct eg:
INPUT"Enter Y/N" answer$
Here "y" is as acceptable as "Y", and "n" is ok as is "N"

*FX202 turns on both Caps lock and Shift lock.
*FX202,0,100 turns on Caps Lock
*FX202,50 turns off both Caps lock and Shift lock.

Another way is by using INKEY minus. Both "Y" and "y" have the same value of INKEY(-69). "N" and "n" have the value INKEY(-86). However the implications of the FX calls above are far wider than this...

****

VDU 26!

Did you realise that VDU26 not only resets text and graphics windows back to their default settings, but also resets graphics origins shifted by the VDU29 command. If you have been busily using VDU29 to place graphics at particular points on the screen with the minimum of fuss, plus used a text window to keep that part intact, it is most disconcerting to use VDU26 to reset X,Y to 0,0 only to have the next CLG has wiped your text window as well!

MOVE TO PRINT

When putting explanatory text alongside a picture or diagram, to look nice it often needs placing with greater precision than that offered by PRINT TAB. eg:

```
10 MODE 2
20 MOVE100,200
30 DRAW700,200
40 DRAW400,500
50 DRAW100,500
60 PRINTTAB(2,27)"triangle"
```

As you can see, the word "triangle" is not centred evenly along the base of the shape. However, you can't do any better than that unless you pretend the text is graphics and you MOVE the word to a more precise position. This is made easy by the VDU5 command. This links the text and graphics cursors so that they move together. VDU4 unlinks the cursors (one for the text window, one for the graphics window).

For the listing above:
In MODE 2 each character takes up 64 points across the x-axis (horizontal, across the screen: 1280points divided by 20 characters = 64). You have an eight-letter word which will therefore take up 8*64 points or 512 points.
The base of the triangle runs from 100 to 700, measuring 600 points across.
The base minus the word length is 600-512=88. These 88 points have to be split evenly to each side of the word - 44 on each side. So to place the word centrally, you have to MOVE to the start of the triangle shape plus 44 to start the word. (MOVE 100+44...) The Y co-ordinate can be fiddled in the same way. So delete line 60 above, and add in these lines:

```
60 VDU5
70 MOVE 144,164
80 PRINT"triangle"
90 VDU4
```

Funny things can happen if you don't separate the cursors (VDU4) at some stage.

triangle

****
stars

triangle

WANTED

NOT-TOO-ADVANCED hardware projects.

MORE listings of Fun, interesting (whose definition?), applications programs.

#,$,#,?
A member has been typing in a machine code listing from a magazine (not ours), but keeps coming across a pound sign. The program doesn't work. On some printers the keyboard pound isn't the printer's pound. Quite often it has the same ASCII value as what on the Beeb produces a hash # symbol. If every pound symbol is replace by a # the program should work (or fall over from some other typo).

NOT WANTED?
Feedback from a member re the latest cover: It makes the mag look more like an ornithological publication! Wouldn't it be more appropriate for graphics (especially on the front cover) to be computer generated.
The cover with index on front was more practical, with index for ready reference.
Ed's REM whose computer is to generate the graphics? What graphics?

HELP GIVEN

Bill Tillick thinks he has discovered how to get the low cost processor, which he presented in the June Beeblet, to work on disc. The suggested amendments are:

WORDPROC

```
450 p%=&1100
590 p%=&1120
```

RWP:

```
280 S%=&1202
620 CALL &1120
710....CALL &1100
2150....CALL &1100.
```

BILL adds, If anyone would like a copy of either the tape or disc version then I can be sent a tape, or a copy can be taken at the Wellington meeting subject to the availability of a machine."

CURSOR HELP

Does anyone know of a way of getting rid of the cursor? Great, everyone, including the User Guide knows that. But everyone also knows that the cursor re-appears when you do a Mode change. (Lloyd expects that, so he's not worried about that either). The cursor also reappears when you activate the COPY cursor - ie press one of the arrow keys when you are INPUTting or GETting a character. What is wanted is a way of getting rid of the cursor and not getting it back, even when the copying cursor has been activated!

REM Don says that VDU23,0,11,0;0;0;0 nicely locks in the cursor for INPUTs. Doesn't solve Lloyd's problem. Something else can though:

```
   10 REM cursor-less input
   20 MODE 4
   30
   40 REM normal input
   50 INPUT"First name",first$
   60
   70 REM input without cursor
   80 PRINTTAB(5,10);
   90 second$=FN_input("second name")
  100
  110 REM print name
  120 PRINTfirst$ SPC1 second$
  130 END
  140
  150 DEF FN_input(prompt$)
  160 LOCAL reply$
  170 char_width%=32:REM this assumes Modes 1 or 4
  180 MOVE (char_width%*POS) , (1023-32*VPOS)
  190 VDU5
  200 PRINTprompt$;
  210 INPUT" ",reply$
  220 VDU4
  230 VDU31,0,VPOS+1
  240 =reply$
```

By using POS and VPOS to find out where on the screen you are, then MOVEing to where the text cursor is, linking the text cursor to the graphics cursor before printing out what would be considered the INPUT statement (prompt$). See also an article titled "Move to Print" in this issue. The program above doesn't interfere with VDU23 cursor removal - that still works before and after using FN_input.

Rudolph & co. swimming to NZ

COMPUTER CENTRE

DREGHORN PLACE TOKOROA

P.O. Box 128 Phone (080)68-922

We are agents for **CLARES** of the United Kingdom.

B-BASE: The best value for money in a database.

REPLICA-II: A Good tape to disc Transfer Disc.

SHADOW: Backup those uncopiable tapes.

THE KEY: Disc Utility. Edit and retrieve discs damaged or erased by a head crash.

JOYSTICK UTILITY: Those programs that only use the Keyboard can now use Joysticks. Works with any program BASIC or M/Code that uses INKEY(-) to detect movement.

As we pass on to the user any savings in Exchange Rates, prices vary therefore listed prices are only a guide.

B-BASE	$89-00	SHADOW	$36-50
REPLICA-II	$49-00	JOYSTICK	$28-55
THE KEY	$52-00		

MAILBAG

More news from Waiheke Island:

Most of you will, I hope, recall John B.'s letter published in the October Beeblet. John has sent me a number of listings from the ELBUG magazine : "DANCE/L", "NIAGRA", "TV", "HARMONICS", "ELBUG", "XMAS", "LUNAR", "FOOT/B", "RUNNING/M", and "SOR/DANCE". The ones I have so far tested run on the BBC also (John - I think all Electron programs will run on the Beeb, however some adjustments may be required to take account of the speed difference). Anyone interested in the listings could send me either a tape (or disc) plus $1-00 p&p plus some contribution for the mag (size doesn't matter), or $3-00 plus mag contribution. For those regular contributors whose tapes I have accumulated, they should be on their way as soon as I have recovered from the combination of silly season and this edition.

Back to John's letter. He assures me, tongue in check, that as he is "not a man of letters.. Beeblet is indeed privileged to have a contribution from me". Well, I am always delighted to get letters, even if they don't come from a "man of letters". John notes that getting rid of ones "junk" is a good idea - the problem is finding a buyer (evidently "one man's junk is another man's joy" does not have an entirely universal application!). He also admits that his Electron was originally destined for a somewhat younger associate of his later this month. Not only has the replacement (a Vic 20) has rather dented his expansion plans, but "I have no doubt that he <the recipient of the Vic 20> will talk me into doing a swap at a later date (much later I hope)".

John would "also be interested to know why people buy computers and to what use they put them (I mean personal computers)." So would I! Judging from the numbers of people who have come to talk to me at work, usually just before they go overseas, a personal computer is THE thing to have after the video recorder / player. However, a little bit of probing (spouse with secretarial or literary skills, involvement in club / society activities, massive record collections, children and "education", a frustrated desire to play arcade games.....) usually prdduces the necessary rationale in short order. Perhaps some of our less cynical members may wish to comment!

FEEDBACK

Neil B. has responded to my plea for a little more feedback..." Can't say I am over keen on the new cover - it makes the magazine look more like an ornithological publication in my opinion. Wouldn't it be more appropriate for graphics (especially on the front cover) to be computer generated?" However, I think that the old cover was more practical, with the index in the front for ready reference." Well Neil, this is precisely what I have been wondering about. Opinion seems divided on this, but so far a clear majority of people who have actually written to me seem to like the new cover. I suppose that people have to actually pick up the magazine and look inside to see what its about - perhaps an advantage where Beeblet is on display in a shop, IF the cover is sufficiently interesting to get people's attention in the first place. I am also trying to create a unique identity for the mag (not necessary the one featured in "Broadsheet"!), however, is it the right identity?

Neil continues, "Another beef now - I'm sorry that I've got very little out of the last two issues - too much chat, reviews and too many utility programs. What I want is more listings of interesting / fun / applications programs and not-too-advanced hardware projects. (gripe, moan... enuff)". Well, I keep saying that the only way to improve the mag is to tell me what you actually want so thanks Neil. Of course, others might like to make some comment as well, before I fill up the mag with listings..... etc. I am sure that everyone will have plenty of time to experiment with their machine over the Christmas break - how about dropping me a line with the results?

Neil wrote his letter using Bill Tillick's version of the wordprocessor from the Computing Today magazine, which he has "finally" managed to adapt to disc (see the help section for Bill's mods.). Neil reports "I can safely report that it is infiately good value for money and I am very pleased with it."

ANOTHER VIEW

Alison writes "More on the September cartoon issue.. This is written by a 30's+ (but not by much) 'GIRL' and I would like you to know that not all women are offended about such issues. The cartoon in question had to be looked at again to make sure nothing had been missed out. I feel that there are far more important issues to get upset over and that women are getting far too touchy on this particular type of subject."

Hmm. Well, I suspect that somewhere between "women" and "52% of the population" there could be between one and a half to two million individually different reactions to the cartoon. Its nice to think that our group has a representative sample...

In reply to your query Alison, the "other Anne's" cartoon has not been published - last month's cartoon, and the one in this edition, were also contributed by Ricardo Delee. If you excuse a borrowing from one of the milder "Irish" (insert the nationality of your choice) jokes, once people can overlook, but twice could cause problems.

Alison and family do appear to appreciate the magazine - she speaks of a "real tussle" for it when it arrives.

* * *

ESCAPE the RETURN

Type in *FX220,13, press return, and see what happens.

RANDOM PALETTE

GCF

```
 10 REM  Flashing stripes
 20 ON ERROR MODE7:STOP
 30 MODE0
 40 VDU5
 50 A%=155
 60 REPEAT
 70   X%=RND
 80   *FX19
 90   CALL&FFF4
100   UNTIL FALSE
110 END
```

Some strange effects can be seen if you attempt to program the ULA palette directly (instead of using VDU19,...); especially when you do it at random as in the example above.

If you want to know what is actually happening, try reading "The Advanced User Guide". When you have understood it, please explain it to me, preferably via an article in Beeblet.

RECURSIVE STRUCTURES

How do you represent a recursive procedure in a structure diagram?

BEEPERS

There once was a lady in Threep,
Never thought SOUND could mean more than BEEP.
'Til FX two-eleven
hit her VDU7;
She's the noisiest lady in Threep.

VDU7 is BEEP but... it can be WHOOP ... or BLIP ... or NOTE ... and all via those FX calls not explained in the U.G. (not OS 0.1)

```
*FX211,n  changes the channel number.
*FX212,   changes the envelope number or amplitude
*FX213,n  changes the pitch
*FX214,n  changes the duration
```

To make VDU7 sound like SOUND1,-15,50,20 set up these definitions:

```
*FX211,1          REM channel 0-3
*FX212,128        REM amplitude -15
*FX213,50         REM pitch = 50
*FX214,20         REM duration = 20
```

The call that needs explaining further is *FX212. The number to put after it is: (amplitude or envelope number minus 1)*8
So to use envelope 2,
*FX212,(2-1)*8 or *FX212,8
for a volume (sorry, amplitude) of -15
*FX212,(-15-1)*8 or *FX212,(16*8) or *FX212,128.

Type these commands into your computer:

```
ENVELOPE1,3,1,1,1,10,10,10,126,0,0,-128,126,126
*FX211,1
*FX212,0
*FX213,105
*FX214,108
```

Now type in VDU7 and listen.
How about

```
ENVELOPE2,10,0,0,0,0,0,0,34,10,-20,-26,100,126
*FX211,0
*FX212,8
*FX213,5
*FX214,50
```

With those two envelopes in your Beeb, type in:
SOUND1,1,105,108 listen... then... SOUND0,2,5,50
sound the same as those Beeps did?

SIDEWAYS ROM

Steve W. Aucland.

Ever wanted to write your own sideways ROM's?. I did, and with the assistance of the October Micro User and the Advanced User guide, I finally succeeded. For testing such software some sort of machine code monitor is essential (I use *ADE*),and of course *SOLIDISK* sideways RAM makes it possible to test the code "in situ", i.e as a ROM. The ROM I created will do a screen dump to a STAR DP515 printer, in any *mode* (text-only in *modes* 3,6 and 7), using the command *DMP. If anyone's interested in getting a copy of this, send an SAE and a cassette to Steve Williams at 61 Disraeli Street, Epsom, Auckland. Send $25 if you want it blown into EPROM (to cover cost of 2764 and copying).

SCREEN DUMP IN BASIC

The following is a procedure which may be useful to include in any program where you want to dump a copy of the screen to a printer. The program revolves around the VDU 31 command (position cursor) and OSBYTE CALL 135 (read character at cursor position).Any non text characters will be printed as a space.

```
  10REM SCREEN DUMP PROCEDURE
  20REM STEVE WILLIAMS
  30REM WORKS IN ANY MODE,TEXT ONLY
  40REM
  50DEFPROCscreendump
  55 LOCAL M%,X%,S%,A%
  60 M%=?&355:REM M%=screen mode
  70 RESTORE 240:REM set data pointer
  80 FOR X%=0 TO M%
  90 READ xmax,ymax:REM read max x,y
 100 NEXT
 110 VDU2:REM switch printer on
 120 FOR suby=0 TO ymax-1
 130 FOR subx=0 TO xmax-1
 140 VDU 31,subx,suby:REM position cursor
 150 A%=135:REM set accumulator
 160 S%=USR(&FFF4) AND &0000FF00:REM osbyte call 135
 170 S%=S%/256:REM pick out X register
 180 IF S%(32 OR S%)127 THEN S%=32:REM convert to space if not text
 190 VDU 1,S%:REM send character to printer
 200 NEXT
 210 VDU1,10:REM end of line,so send linefeed
 220 NEXT
```

```
  230 VDU3:REM switch printer off
  240 @DATA
80,32,40,32,20,32,80,25,40,32,20,32,40,25,40,25
  250 ENDPROC
```

LORDS OF TIME - Level 9

The masters of Adventure games have done it again. Having spent most of my spare time for the last two weeks playing this game, I'm finally getting close to the solution (still only 825 out of 1000 though). If you only ever play one Adventure game, then play one from Level 9, and please BUY a copy, we want to keep these guys in business!. I don't think these games are available in NZ, so you'll have to send to UK (229 Hughenden Road, High Wycombe, Bucks, United Kingdom).

For anyone stuck in Colossal Cave, Snowball, or Lords of Time I can offer assistance (please send an SAE).

GIVE-AWAYS

Tony W in Napier advises us that those who have read The Micro User recently will know that a new factory in Dublin will soon be producing BBC Micros. What they may not know however, is that with each micro there is a free gift. Don't rush, says Tony - it's a dozen bottles of Twink!!

* * *

GCOLERIC

There once was a colourful beastie,
Went out for a walk in GCOL3
His steps out he retraced,
To come home. We're amazed!
He's invisible that beastie in GCOL3!

```
 10 MODE2
 20 VDU5
 30 GCOL3,1
 40 FOR I% = 0 TO 1000 STEP 100
 50   MOVE I%,100
 60   PRINT"B"
 70   wait=INKEY(50)
 80   NEXT I%
 90 FOR I% = 1000 TO 0 STEP -100
100   MOVE I%,100
110   PRINT"B"
120   wait=INKEY(50)
130   NEXT I%
140 VDU4
150 END
```

INVALUABLE UTILITIES FOR THE BBC MICRO.

JEFF AUGHTON PAN BOOKS

Reviewed by James Logie, Auckland.

The title was enough to arouse my interest and after as long as could be decently spent looking at it without actually becoming a permanent fixture I decided that I would buy the book. After owning it for over a month now I thought that it might be helpful to let others know what I thought of it.

Most experienced and/or wealthy micro owners will have either a ROM full of utilities or will have written their own routines to cope with the problems arising when writing programs of their own, trying to get their masterpiece functioning, or attempting to get other people's efforts to work as expected. Being a poor novice hoping to write programs of his own this seemed a cheap way to learn while getting a useful return for my outlay. As the author hints in his book every Tom, Dick and Harriet has their own favourite way of doing things and utilities abound, while not as yet in the dime a dozen class (ROM based anyway). The monthly magazines usually have some sort of routine or utility so what advantage if any does this book have over what else is around ? In its 236 pages there are 31 utilities. They range from a Block Memory Move on a user defined function key to a 300 line Music Processor; with most of the rest being less than half that length. It seems pointless to make a list of all the routines and for various reasons I have not tried them all.

For each utility the author gives a description of the program, a how to use it section and a listing (in a large easy to read type). Then follow how it works, what the procedures do and the variables used. A final section on extensions gives suggestions as to how the program could be modified or extended with in some cases a practical example. However with the amount of information given extension or alteration of the main listing should not prove too difficult, and this for me was one of the main reasons for buying a do it yourself utility kit in the first place. The author seems to have found a good balance between assuming that the reader does not know very much about computing but is not completely ignorant.

Having recently bought a disc drive I have made good use of a utility to download the programs after loading from disc, and have also used one that allows direct access to the disc to examine and edit, if required, the information stored there. As the author points out it is a facility to be used with care.. but invaluable if you have just accidentally *WIPEd a program or have other information there which otherwise you have no access to.

A similar program allows viewing and alteration of the memory in the machine. It is interesting just to scroll through the memory without attempting to alter anything. See how Acorn write BASIC. See what you have just written. See yourself looking at what you have just written. For writing programs there is a Graphics utility which allows graphics to be drawn on the screen, colours changed prior to and during drawing and a record continuously presented of the current position. A Character Creator 24x8(pixels), can colour each individual pixel in full Mode 2 colours, show the creation in all its glory, allow alteration, show the VDU statements necessary to program it and also save it.

Other utilities assist in writing programs, replacing strings, condensing programs, screen save and printer memory dump.

I have had a lot of fun using some of these routines, have learnt something about the BBC micro and intend to make more use of the programs I now have and those that I have not as yet attempted. When I become more familiar with what I am doing I expect to be using the suggestions in the book to extend and modify the programs to suit my needs.

To sum up, here for the price of a cassette game, in a readable book are 30 plus programs that could amuse, assist and perhaps inspire you in your conflict with your micro.

P.S. This review was written with the aid of Bill T's "cheap" wordprocessor. Thank you Bill T. (It loaded perfectly into Wordwise too! -Ed.)

TROUBLED ADVENTURER

Diarmid F. would like to know "how to stop the Zombie from wasting me when getting the trident ie. where are the hammer and nails."

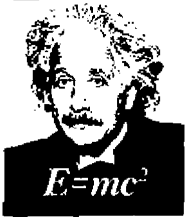
I understand that it might help if you explore the area between the blobs of jelly and the yellow room more thoroughly. On the other hand, it may not! Happy adventuring - Ed.

EINSTEIN SCIENTIFIC LIMITED
177 Willis Street
WELLINGTON.

CHECK EINSTEIN'S EXTENSIVE RANGE OF PRINTERS AND SOFTWARE FOR THE BBC & ELECTRON MICROS.

WELLINGTON
177 Willis Street. Phone Raju, Michael or Malcolm 851055 or 844353.
PALMERSTON NORTH
154 Broadway Avenue. Phone Richard 64108.
NEW PLYMOUTH
Corner King & Egmont Streets. Phone Bill or Sally 85528.
CHRISTCHURCH
Shop 41, 1st floor, Cashfields Mall Phone Mike 66442.
HAMILTON
inside DIC, Garden Place. Phone Maxine 81969.

Einstein
SCIENTIFIC LTD



The computer world made simple.

THANKS FOR THE MEMORY

David Fung

To the new and prospective members of the user group: A most important reason why you should join the group is that by talking to other members about your problems, helpful hints, clues and flashes of brilliance often restore your sanity and happy relationship with your Beeb/Acorn. The other advice I have is to set yourself a definitive task, it would not matter how simple it is, but persist with it until the end. You will be pleasantly surprised how much you do learn in the process.

I had little idea about computer and programming when I joined the group. I had the crazy notion that I would like to develop a Chinese Word Processor. Thanks to the tips and advice from the other members like Gavin, Anton, Neil, Steven and others I have achieved my objective in a much shorter time and more pleasurable then otherwise.

One of the major problem I had was the limited memory in the Beeb and externally in the discs. This is common to any applications using indices and files. I would like to share my experience with the not so experienced members in recognition to the help and encouragement given to me so generously by the OLDIES.

STRINGS$ HAVE STRINGS ATTACHED

If you want to set up an index of say 1000 names and their corresponding record numbers, the simplest way is to reserve space by the DIM command, ie, DIM name$(1000), record%(1000). Let's say the names are each 10 characters long. You would need 11 bytes for each name$. Each record% would also need 4 bytes. So you would need

1000 x 11 + 1000 x 4 = 15000 bytes

The computer has to keep track of where these strings are stored so it sets up its own index of STRING INFORMATION BLOCKS and there is one for each string. Each block uses 4 bytes. So the computer grabs

1000 x 4 + 1000 x 4 = 8000 bytes

You have used up 15000 + 8000 = 23000 bytes or 22.46K of memory. This will leave you with about 2.29K in mode 7 which is the only mode you can work in.

The way to get around this is to do away with STRINGS $/%

Let's reserve 1000 x 10 bytes for the names and we can use 2 bytes for each record number if the number is smaller then 65536. The total memory requirement is 1000 x 10 + 1000 x 2 = 12000 bytes made up of 1000 blocks of 12 bytes.

Now go to your computer and switch it on (if it was on, switch it off then on.) You are now in mode 7.

Type `P.~( HIMEM - 12000),~HIMEM`

The computer will say `4D20   7C00`

Type `HIMEM = &4D1F`

The memory location between 4D20 and 7C00 is now reserved for your index. You can *SAVE this as an index file by type in

```
*SAVE INDEX  4C20 7C00
```

The file contains 12000 zeros at the moment.

You can now start writing to your index with the following programme

```
60 record% = 1
80 begin% = &4C22        REM. begining of index
130 INPUT'" PLEASE ENTER NAME ",name$
170 $begin% = name$      REM. copy name$ there
180 begin%?10 = record% MOD 256
190 begin%?11 = record% DIV 256
```

begin%   is the label for the first byte where you start to write a name in the index
begin%?10 is the 10th byte from the first one therefor the 11th byte, in which you put the REMAINDER of (record% divides by 256)
begin%?11 is therefor the next, the 12th byte for the WHOLE NUMBER of (record% divided by 256)

To make sure that your names are not longer then 10 characters add the following lines:

```
120 REPEAT
```

```
140 IF LEN(name$) > 10 PRINT'"NAME TOO LONG"
150 UNTIL LEN(name$) < 11 AND LEN(name$) > 0
```

To make this operation automatically repeat until you want to quit type these in:

```
100 REPEAT
200 begin%=begin%+12      REM. next block of 12
210 record% = record% + 1 REM. next record number
240 UNTIL record% = 1001
250 PRINT'''TAB(10)" INDEX IS FULL"
```

You can quit by pressing ESCAPE but on RUN again you would not know where you were up to and the programme would start from the beginning with record 1. So you will need to write down somewhere where you were, not on a peice of paper, but in your index, say in the first two bytes of the index?

```
40 ON ERROR ?&4D20 = record% MOD 256 : ?&4D21 = record%
DIV 256
60 record% = ?&4D20 + ?&4D21 * 256      REM. reads next
record%
80 begin% = &4D22 + (record% - 1) *12      REM. works out
next block
90 IF record% = 1001 THEN PRINT " INDEX IS FULL" : END
```

You would also want to save whatever you have put in the index, so:

```
40 ON ERROR  PROCsavedata
270:
280 DEF PROCsavedata
290 ?M = record% MOD 256 : M?1 = record% DIV 256
300 REPEAT
310 INPUT'"DO YOU WISH TO SAVE INDEX        "A$
320 UNTIL A$ = "Y" OR A$ = "N"
330 IF A$ = "N" THEN END
340 *SAVE INDEX 4D20 7C00
350 END
360 ENDPROC
```

Now List your program and Bingo you have an " INDEX MAKER ", an INDEX of 1000 blocks of 12 bytes and you still have 13K of memory for you to access "STRUCTURALLY".

Type in the following to complete the programme and to make it more friendly and to tidy up the loose ends, then save the whole listing under a file name eg. "I.MAKER"

```
10 MODE 7
20 P.TAB(10,10); "INDEX MAKER"
30 WAIT = INKEY$(200)
50 M = &4D20
70 IF record% = 0 THEN record% = 1
110 CLS
160 PRINT' record%," ",name$
220 PRINT''" PRESS ANY KEY TO CONTINUE "
230 KEYHIT=GET
260 END
```

Now run the program and put in some names. If you want to leave off and come back next day to put more names in, it would make life easier if the programme would *LOAD the index for you and reset HIMEM at the same time. Also you can sneak in an extra 1.25K memory if you reset PAGE from &1900 to &1400. You can create a "LEAD" to do that and "CHAIN" "I.MAKER" for you.

```
10 REM. "LEAD" to chain "I.MAKER"
20 HIMEM = &4D1F
30 PAGE =&1400
40 *LOAD INDEX
50 CH. "I.MAKER"
```

Save that under the file name "LEAD"

With the indulgence of the editor I shall in subsquent articles show you how to read your index, how to sort the names in alphabetical order, how to insert a name in the already sorted index and how to write to and read from the files related to the index in the most memory sparing ways.

Meantime why don't you set yourself a task to beat me to it. I am sure the editor would be please to receive your effort. Perhaps he may even offer a prize for the best masterpiece!

Hint:

You can have your cake and eat it too. If you use an extra byte for your name and poke 13 into it then you can manipulate the names as name$s and get away with no STRING INFORMATIOM BLOCKS because you should know where the name$s are.

If you are confused then you should start asking questions. That is what the user group is for.

Best wishes!

CLASSIFIED ADS

FOR SALE:
1 MPI Disc Drive
80 Tracks Double Sided
Has recently been completely overhauled.
Fully cased and all cables included
Price $850-00 ONO
Write to John 49 Coronation Road, Morrinsville
Or phone MOV 6695 (after 5pm)

FLOPPY DISC CONTROLLER

Is there anyone with a 8271 FLOPPY DISC CONTROLLER which they do not need, or who knows where on can be obtained in N.Z. Willing to pay up to $120-00.
Please call PATRICK at 367-722 or write to

PATRICK DUGHTON,
4 PIKARERE STREET,
TITAHI BAY,
WELLINGTON

MICRO COVERS

Designed to fit the BBC micro. Attractive opaque reinforced plastic, $5-50 posted. To Iain, Box 26078, Epsom, Auckland 3.

DEALS AND DISCOUNTS

Send stamped addressed envelope for details of the deals negotiated by the Wellington Branch for UG members. Please mark your request for the attention of Jill Armstrong, C/O the UG Box 9592 Wgtn.

CASSETTE TAPES

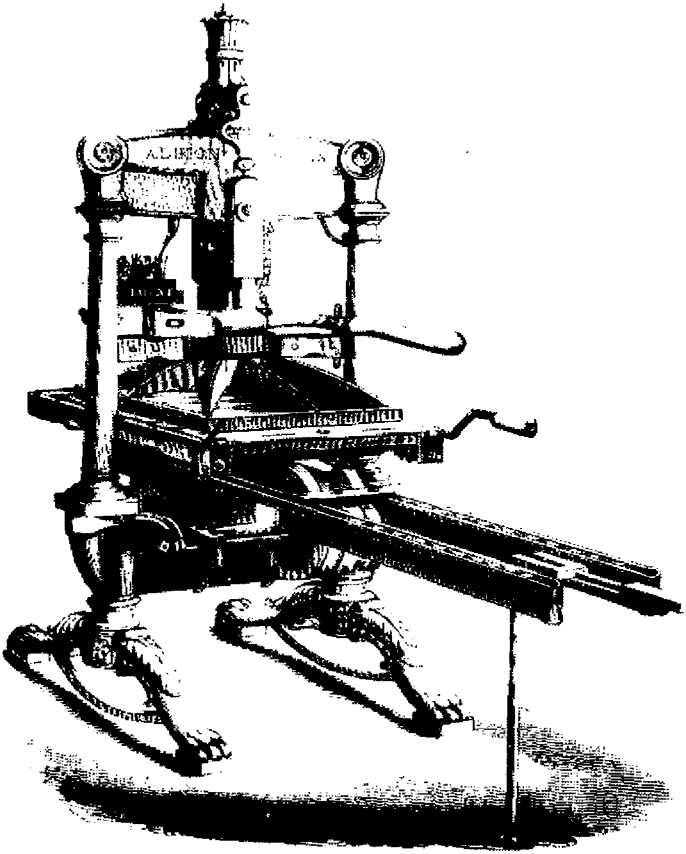
C10s, boxed, leaderless - available from the UG in multiples of 20 : $26-00 plus $2-00 p&p.

FLOPPY DISCS

3M Brand 5 1/4" DSDD discs - available from the UG ex-stock in (cardboard) boxes of 10: $55-00, which includes p&p.

...AND ONE FOR THE ROAD...

Do we have any FIAT fanatics in the group? Well, the Editor has this very quick 124 1600 Sports for sale...



Roberts Printing Co. Ltd.

Telephone 846 472 850 831 P.O. Box 6074 WALTER STREET, WELLINGTON

(Off Vivian Street between Cuba and Willis Streets)

HALL OF FAME

PUBLISHER:	GAME	HISCORE	SCORER
Acorn	Aviator	150	Kendal
	Countdown	227	Tim R.
	Elite	Average	K. Finlay
	Hopper	27,196	Jenny B.
	Meteors	24,090	Duncan's flatmate
	Monsters	93,560	Linsay F.
	Starship Command	5,497	Robert N.
A&F	Chuckie Egg	324,210	Mike D.
A&F	Frenzy	807,918	Mike D.
A&F	Cylon Attack	81,600	Robert N.
Alligat.	Bug Blaster	37,642	Steve W.
Amcom	Fortress	68,000	Joanne C.
Badger	Lunar Lander	245,967	Duncan's flatmate
Dr. Soft	'747'	205	Steve W.
	Gorf	18,648	Steve W.
FG Soft.	Mutant Defender	6,900	Steve W.
G.C.F.	Mental Arithmetic	240	Gavin F.
Level 9	Colossal Cave	687	Steve W.
	Dungeon Adv.	330	Anne et al
Prog.Pow	DangerUXB(Lvl 1)	546,990	Mike's daughter
	Felix the Factory	7,000	Dianne S.
Sup.Soft	Alien Dropout	24,200	David E.
	Q*BERT	30,260	Dianne S.
	Roadrunner	70,600	Lois S.

HALL OF FAME

USER GROUP MEETINGS

AUCKLAND:-meets 2nd Wednesday of the month, 7.30pm, Conference Rooms,
Auckland Teachers College,60 Epsom Ave,Epsom
ph Dave 770-630 x 518(bus) or Kerry 695-355.

CHRISTCHURCH:-fortnightly workshops, Monday 6.30pm at Hagley High School. ph. Michael 582-267.

HAMILTON:-last Thursday of the month 4.45pm Waikato Tech B-block staffroom. Ph. Peter, Hamilton 392-500 x 897(bus) or Alison, Morrinsville 6695.

HAWKES BAY:-Alternative months Hastings & Napier. Contact Kendall Napier 435-624, Bob Taradale 446-955, Mitch Hastings 778-235.

INVERCARGILL:-Phone Evan 330-444 Invercargill

Palmerston North:-Phone Tom Skinner 83-009 (wk)

TAUPO:-Contact Dave 84-215.

TAURANGA:-every second Wednesday. ph. Chris & Jane 65-076.

TIMARU:- Meetings to be arranged in the first week of each month. Contact Chris Stone ph. Timaru 60394

WELLINGTON:-meets LAST Thursday of the month, 7.30pm, Correspondence School staffroom (1st floor), Portland Cres, Thorndon. ph. Anton 286-289 or Warren 787-005.

OTHER CENTRES:- let us know and we will publish details here.

THE NEWSLETTER:

CONTRIBUTIONS:-most welcome. Listings should be sent in on tape or disc.(Tape - 2 copies,one at 300 baud please). A 35c stamp gets your tape / disc returned too. Please include written explanation of listing; text only material welcome too.

DEADLINES:- For a particular issue, the last day for material is the last day of the month prior to publication.

ADVERTISING:- Rates are $20 per half page, deadline as above.

Copyright (c) THE BBC/ACORN COMPUTER USER GROUP OF NZ INC.

Published monthly except January and mailed to financial members.

MEMBERSHIP:

Membership of the users group is on payment of an annual subscription from April to March. For the 1984/5 year it is: $30.00, plus a $5.00 joining fee for new members, plus $5.00 for Auckland & Wellington members as a local branch levy, which entitles attendance at meetings as detailed above. Renewing members get all back issues of the current year; New members also get back issues to v.1 #1 Jan '84.

THE ACORN/BBC COMPUTER USER GROUP OF NZ INC
P.O.Box 9592
Wellington

Reproduced by ROBERTS PRINTING Co. Ltd, Wellington.